REAR ECU DRIVERLESS CAR

A Degree Thesis

Submitted to the Faculty of the

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

by

Quim Pérez Jiménez

In partial fulfilment

of the requirements for the degree in

TELECOMMUNICATION TECHONOGIES AND SERVICES ENGINEERING

Advisor: Fernando Silva Martínez

Barcelona, June 2020

Abstract

Formula Student is a worldwide competition where students manufacture formula race cars. Last year, the Driverless UPC team, formed by ETSEIB and ETSETB students, adapted all the electronic systems of a classic electric formula car, to transform it in an autonomous car. The result was a mix of the autonomous part and the old electrical car electronics causing the reliability of the electronics was significantly low.

For this reason, the aim for this year was to join all electronics and manufacture a robust single electronic package. This thesis is focused in the development and designing of the Rear ECU and DC-DC regulators implemented in all ECUs. The results are a compact and functional Rear ECU, and high-efficient with low overheating DC-DC regulators. This leap to the conclusion that joins various functionalities inside an ECU and designing your own voltage regulator is highly recommended for a reliable and efficient electronics.

Resum

Formula Student és una competició mundial on els estudiants fabriquen cotxes estil fórmula. L'any passat, l'equip Driverless UPC, format per estudiants ETSEIB i ETSETB, va adaptar tots els sistemes electrònics d'un cotxe de fórmula elèctrica clàssica, per transformar-lo en un cotxe autònom. El resultat va ser una barreja de la part autònoma i l'antiga electrònica del cotxe elèctric provocant que la fiabilitat de l'electrònica fos significativament baixa.

Per aquest motiu, l'objectiu d'aquest any era unir tota la electrònica en una única i robusta unitat electrònica. Aquesta tesi està enfocada al desenvolupament i el disseny de la Rear ECU i els reguladors DC-DC implementats a totes les ECU. Els resultats són una Rear ECU compacta i funcional, i uns reguladors de tensió d'alta eficiència i sobreescalfament baix. Això porta a la conclusió que unir diverses funcionalitats dins d'una ECU i dissenyar un regulador de tensió propi és altament recomanable per a una electrònica fiable i eficaç.

Resumen

Formula Student es una competencia mundial en la que los estudiantes fabrican autos de carreras fórmula. El año pasado, el equipo Driverless UPC, formado por estudiantes de ETSEIB y ETSETB, adaptó los sistemas electrónicos de un clásico automóvil fórmula eléctrico, para transformarlo en un automóvil autónomo. Esto resultó una mezcla de la parte autónoma y la vieja electrónica del automóvil causando una fiabilidad electrónica significativamente baja.

Por esta razón, el objetivo para este año era unir toda la electrónica en una única y robusta unidad electrónica. Esta tesis se centra en el desarrollo y diseño de la Rear ECU y los reguladores DC-DC implementados en todas las ECUs. Los resultados son una ECU trasera compacta y funcional, y reguladores DC-DC de alta eficiencia y sobrecalentamiento bajo. Esto lleva a la conclusión de que unir varias funcionalidades dentro de una ECU y el diseño de un regulador de voltaje propio es altamente recomendable para una electrónica confiable y eficiente.

Dedicated to all my family and especially to my parents and my sister who raise me and educate to be the man I am today. Also, a special mention for my father's uncle, Juan Sisinio, who inspires me to study and become the best version of myself.

Acknowledgements

I wish to express my sincere thanks to all the team members of the Driverless UPC project, in particular to the team leaders and my section chief, who allowed me to take part in this amazing project and have made all this possible.

I also thank to my Project Supervisor, Ferran Silva, to guide me through my thesis.

And finally, I place on record, my sense of gratitude to one and all who, directly or indirectly, have lent their helping hand in this venture.

Revision history and approval record

Revision	Date	Purpose
0	22/03/2020	Document creation
1	16/06/2020	Document revision

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Quim Pérez Jiménez	quimperezj98@gmail.com
Fernando Silva Martínez	ferran.silva@upc.edu

Written by:		Reviewed and approved by:	
Date	14/06/2020	Date	16/06/2020
Name	Quim Pérez Jiménez	Name	Fernando Silva Martínez
Position	Project Author	Position	Project Supervisor

Table of contents

The table of contents must be detailed. Each chapter and main section in the thesis must be listed in the “Table of Contents” and each must be given a page number for the location of a particular text.

List of Figures

List of Tables:

1. Introduction

Before talking about the project itself, let's introduce what the Driverless UPC consists on and what Formula Student is. To do that, Formula Student Germany (FSG), the best competition inside Formula Student, defines the project as:

[1] "*Students build a single seat formula race car with which they can compete against teams from all over the world. The competition is not won solely by the team with the fastest car, but rather by the team with the best overall package of construction, performance, and financial and sales planning.*"

"*Formula Student challenges the team members to go the extra step in their education by incorporating into it intensive experience in building and manufacturing as well as considering the economic aspects of the automotive industry. Teams take on the assumption that they are a manufacturer developing a prototype to be evaluated for production."*

"*The challenge the teams face is to compose a complete package consisting of a well-constructed race car and a sales plan that best matches these given criteria.*"

"*The jury will judge every team's car and sales plan based on construction, cost planning and sales presentation. The rest of the judging will be done out on the track, where the students demonstrate in a number of performance tests how well their self-built race cars fare in their true environment.*"

Figure 1.1: Formula Student Germany 2018

Driverless UPC was founded two years ago with a clear objective in mind: create the first Autonomous Formula Student Car in Spain. In its first year, there wasn't any kind of background, which implied a real challenge because a lot of work had to be made from scratch. However, thanks to all the team members' effort the car was able to complete all dynamics event at the last competition in Spain, being one of the fewest cars to success in its first year. The car ran, but it did slow (only 10 km/h) and the electrical package had

some serious problems which forced the team to stop the car several times due to electrical or mechanical failures and fix them.

Taking all these into account, the team members set the goal of this second year: the car should be electrically and mechanically more reliable and faster. Having a more robust electrical system would allow the team to test during a larger period of time the autonomous systems and, as a result, the team would have more time to balance and debug all the control and perception algorithms. This fact would enable us to reach higher speeds at dynamics events. Therefore, the current Hardware and Electronics department work is to gather all the designs that were developed the previous year to improve them and correct all the failures that were made due to the lack of time.

Figure 1.2: Driverless UPC car, Xaloc, calibrating Perception system

In particular, this thesis describes the design and development of one of the main Electronic Control Units (ECU), the Rear ECU, and the design of more efficient Direct Current to Direct Current (DC-DC) voltage regulators to adapt the voltage from the Low Voltage (LV) battery to the voltage required on the ECU.

1.1. Objectives

Following the purpose of the project and the thesis focus, the project main goals are two:

1. - Design and manufacture of the Rear ECU.
2. - Design of the DC-DC regulators installed in all ECUs of the car.

The Rear ECU, previously called LV ECU, is in charge of controlling the following systems: the brake light, the enable of the inverter, the cooling system of the accumulator, inverter and motors, and supplies the electronics inside the accumulator container and the Tractive System Active Light (TSAL). In addition, it acquires the rear wheels speed and it senses some points of the Shutdown Circuit (SDC) for telemetry usages. This ECU had to be redesigned to accomplish the new requirements and specifications of the new

car model. This entails a new Printed Circuit Board (PCB) design, the soldering of the PCB, program the microcontroller to implement all the required functionalities of this ECU and the final test of the electronic unit.

Apart from that, the last season DC-DC regulators were significantly inefficient and suffer a meaningful overheating. As a result, the team decided to invest in the design of new DC-DC regulators formed by commercial DC-DC controllers which would give more freedom in the design of the entire regulator and would improve its overall performance.

1.2. Requirements and specifications

1.2.1. Requirements

Project requirements:

- DC-DC Regulators must supply a constant voltage and the required current with at high efficiency.
- The Rear ECU must be able to communicate with other ECUs of the car and the dSpace through the CAN bus.
- The Rear ECU must have an auto diagnosis protocol to inform about the condition of different elements inside or outside the ECU (for telemetry usage).
- The Rear ECU must be able to control the brake light and enable the inverter when a specific CAN message is received.
- The Rear ECU should be able to control the cooling system depending on state of the car and CAN message sent by the dSpace.
- The Rear ECU must be able to supply the TSAL and the electronics inside the accumulator container.
- The Rear ECU must be able to get the data supplied by the wheels speed sensor installed at the rear wheels and transmit this information to the dSpace so it can calculate the speed of the car and use this variable for the control of the car.
- Minimize CAN bus occupation (CAN messages optimization).

1.2.2. Specifications

Project specifications:

- Minimum efficiency of the DC-DC regulators: 95%.
- Low power dissipation in transistors: <1W.
- Maximum current supplied by the DC-DC regulators: 10A.
- Maximum peak current supplied by the DC-DC regulators: 6A.

- Low voltage ripple of the DC-DC regulators: <5%.
- Low voltage fall when we demand more power: <5%.
- Achieve a speed sensibility of 0.1 m/s (around 10 m/s).

1.3. Work plan

1.3.1. Work Breakdown Structure

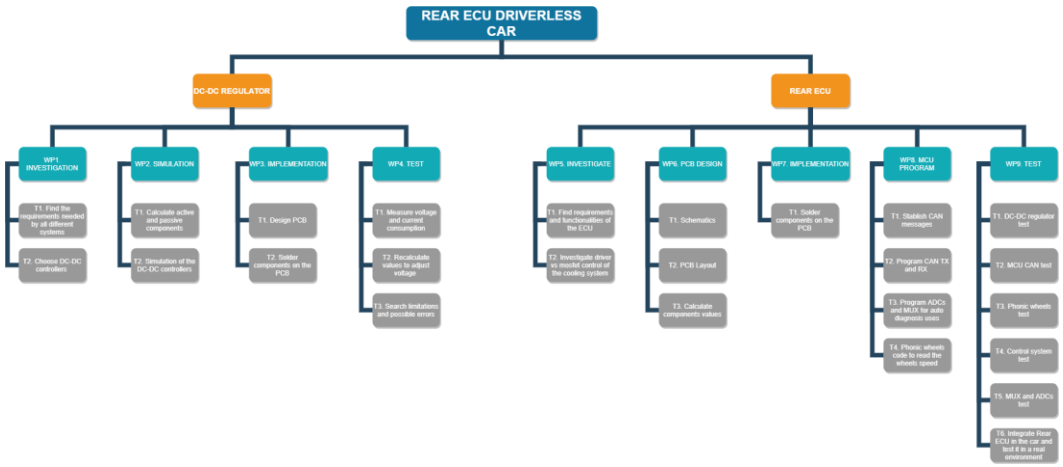


Figure 1.3: Work Breakdown Structure Diagram

The Work Plan is divided in two main projects:

- DC-DC REGULATOR: First, it entails an investigation phase to find the requirements of all ECUs and explore all possibilities for the design of the regulator. After that, simulations were carried out to check the correct functioning of the DC-DC controllers chose. Next, a design of a PCB was needed to test the new voltage controllers. Finally, the DC-DC regulators were tested.
- Rear ECU: First, it entails an investigation phase to find the requirements and functionalities of the ECU. Apart from that an investigation of the control of the cooling system accomplished. Then, the Rear ECU PCB was designed and all components used in this ECU were calculated and chose. After that, all these components were soldered on the PCB. Next, the microcontroller was programmed. Finally, all the features and functionalities of the ECU were tested to check the correct functioning of the PCB.

1.3.2. Work packages, Tasks and Milestones

Project: DC-DC Regulator	WP ref: WP1
Major constituent: Investigation	Sheet 1 of 9

Short description: Investigate the requirements of the different ECUs to choose the best possible DC-DC controller that could be integrated satisfactorily.	Planned start date: 15/09/2019 Planned end date: 1/11/2019	
	Start event: 15/09/2019 End event: 24/11/2019	
Internal task T1: Find the requirements needed by all different systems Internal task T2: Choose DC-DC controllers	Deliverables: List of chosen DC-DC controller	Dates: 24/11/2019

Table 1.1: WP1

Project: DC-DC Regulator	WP ref: WP2	
Major constituent: Simulation	Sheet 2 of 9	
Short description: Calculate the passive and active components of the DC-DC regulator and simulate its response and performance.	Planned start date: 1/11/2019 Planned end date: 28/12/2019	
	Start event: 1/11/2019 End event: 28/12/2019	
Internal task T1: Calculate active and passive components Internal task T2: Simulation of the DC-DC controllers	Deliverables: List of chosen components & temporal simulation	Dates: 8/12/2019 28/12/2019

Table 1.2: WP2

Project: DC-DC Regulator	WP ref: WP3	
Major constituent: Implementation	Sheet 3 of 9	
Short description: Design PCB and solder the chosen components of the DC-DC Regulator on it.	Planned start date: 25/11/2019 Planned end date: 16/02/2020	
	Start event: 25/11/2019 End event: 16/02/2020	
Internal task T1: Design PCB Internal task T2: Solder components on the PCB	Deliverables: PCB layout &	Dates: 8/12/2019 16/02/2020

	Soldered PCB	

Table 1.3: WP3

Project: DC-DC Regulator	WP ref: WP4	
Major constituent: Test	Sheet 4 of 9	
Short description: Test of the DC-DC Regulator. Search its limitation and check simulations results.	Planned start date: 17/02/2020 Planned end date: 02/02/2020	
	Start event: 17/02/2020 End event: 16/03/2020	
Internal task T1: Measure voltage and current consumption Internal task T2: Recalculate values to adjust voltage Internal task T3: Search limitations and possible errors	Deliverables: Final DC-DC Regulator	Dates: 21/03/2020

Table 1.4: WP4

Project: Rear ECU	WP ref: WP5	
Major constituent: Investigation	Sheet 5 of 9	
Short description: Investigate about requirements and functionalities of the Rear ECU and other related aspects.	Planned start date: 15/09/2019 Planned end date: 26/10/2019	
	Start event: 15/09/2019 End event: 26/10/2019	
Internal task T1: Find requirements and functionalities of the ECU Internal task T2: Investigate driver vs MOSfet control of the cooling system	Deliverables: I/O and connections list of the Rear ECU	Dates: 29/10/2019

Table 1.5: WP5

Project: Rear ECU	WP ref: WP6
Major constituent: PCB Design	Sheet 6 of 9
Short description: Design of schematics and PCB layout.	Planned start date: 7/10/2019 Planned end date: 17/12/2019

	Start event: 7/10/2019 End event: 17/12/2019	
Internal task T1: Schematics Internal task T2: PCB Layout Internal task T3: Calculate components values	Deliverables: Rear ECU Schematics, Rear ECU PCB layout and components list	Dates: 27/10/2019 8/12/2019 21/12/2019

Table 1.6: WP6

Project: Rear ECU	WP ref: WP7	
Major constituent: Implementation	Sheet 7 of 9	
Short description: Solder the chosen components for the Rear ECU.	Planned start date: 26/01/2020 Planned end date: 22/02/2020	
	Start event: 26/01/2020 End event: 23/02/2020	
Internal task T1: Solder components on the PCB	Deliverables: Soldered Rear ECU PCB	Dates: 23/02/2020

Table 1.7: WP7

Project: Rear ECU	WP ref: WP8	
Major constituent: MCU program	Sheet 8 of 9	
Short description: Program the microcontroller installed on the Rear ECU to accomplish its functionalities	Planned start date: 16/12/2019 Planned end date: 14/03/2020	
	Start event: 16/12/2019 End event: 14/03/2020	
Internal task T1: Stablish CAN messages Internal task T2: Program CAN TX and RX Internal task T3: Program ADCs and MUX for auto diagnosis uses	Deliverables: Microcontroller code	Dates: 15/03/2020

Internal task T4: Phonic wheels code to read the wheels speed		

Table 1.8: WP8

Project: Rear ECU	WP ref: WP9	
Major constituent: Test	Sheet 9 of 9	
Short description: Rear ECU test.	Planned start date: 10/02/2020 Planned end date: 12/04/2020	
	Start event: 10/02/2020 End event: 12/04/2020	
Internal task T1: DC-DC regulator test Internal task T2: MCU CAN test Internal task T3: Phonic wheels test Internal task T4: Control system test Internal task T5: MUX and ADCs test Internal task T6: Integrate Rear ECU in the car and test it in a real environment	Deliverables: Final Rear ECU PCB with all functionalities implemented	Dates: 12/04/2020

Table 1.9: WP9

Milestones

WP#	Task#	Short title	Milestone / deliverable	Date (week)
WP1	T2	Choose DC-DC Controller	List of chosen DC-DC controllers	24/11/2019
WP2	T1	Calculate active and passive components	List of chosen components	8/12/2019
WP2	T2	Simulation of the DC-DC controllers	Temporal simulation	28/12/2019
WP3	T1	DC-DC Regulator PCB design	DC-DC Regulator PCB layout	8/12/2019
WP3	T2	Solder Components on the DC-DC Regulators	Soldered DC-DC Regulator PCB	16/02/2020
WP4	T1,T2,T3	Test DC-DC Regulators	Final DC-DC Regulator	21/03/2020
WP5	T1	Find requirements and functionalities of the ECU	I/O and connections list of the Rear ECU	29/10/2019
WP6	T1	Schematics	Rear ECU schematics	27/10/2019

WP6	T2	PCB Layout	Rear ECU PCB layout	8/12/2019
WP6	T3	Calculate components values	List of components of the Rear ECU	21/12/2019
WP7	T1	Solder components on the PCB	Soldered Rear ECU PCB	23/02/2020
WP8	T2,T3,T4	Microcontroller code	MCU Program	15/03/2020
WP9	T1,T2,T3, T4,T5, T6	Rear ECU Test	Final Rear ECU PCB	12/04/2020

Table 1.10: Milestones

1.3.3. Gantt Diagram

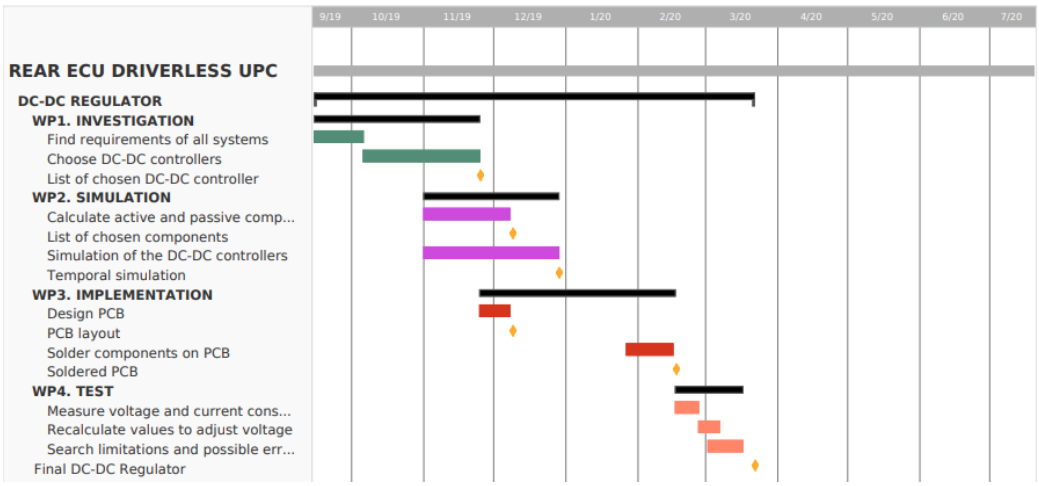


Figure 1.4: Gantt Diagram DC-DC Regulators

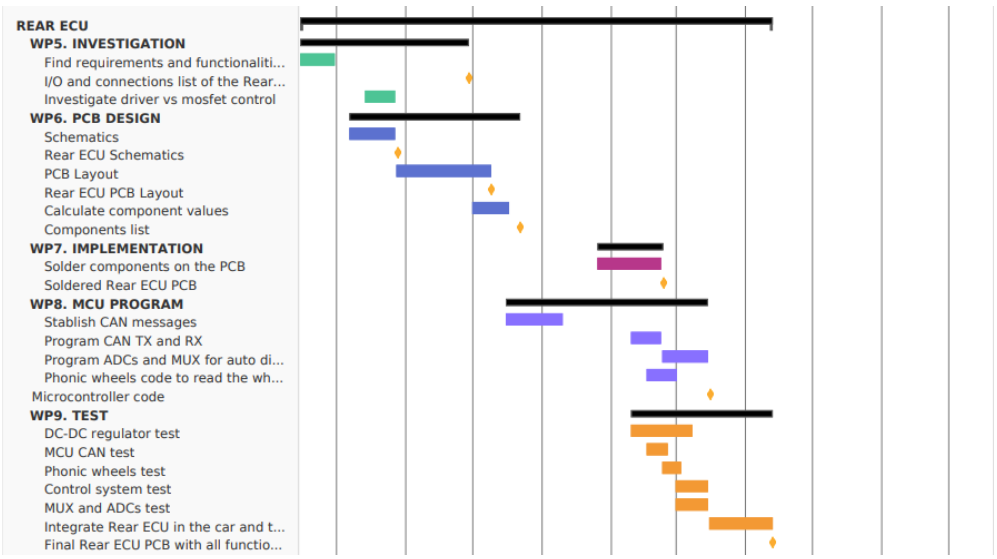


Figure 1.5: Gantt Diagram Rear ECU

Figure 1.6: Gantt Diagram Competitions

Apart from the tasks and WPs already mentioned in the work breakdown structure, also, it is taken into account the competition preparation and the competition itself as part of the Gantt Diagram, because is the main test of the developed systems, its importance for the whole project and the vast time dedicated to it.

1.3.4. Meeting and communication Plan

Inside the project we had two types of meetings: the section meetings, in this case the electronic meetings, and the team meetings.

- Electronic meetings: In these meetings, we exposed what we had been developing during the week, the possible improvements and the distribution of tasks. They were programmed once a week.
- Team meetings: these meetings were used to discuss about the current state of the project and general aspects of the competition, and stablish crucial deadline. They were programmed, at least, once a month or when an important event concerned the whole team.

Talking now about the communication plan of the thesis, it was planned to meet with the project supervisor once a week on Thursdays to discuss about the state of the project and inform about the work done within it; but, due to the COVID-19 outbreak and the resulting measures taken by the Spanish government, the meeting plan was reduced to share information about the project via email.

1.4. Deviations from initial plan and incidences

During the project there were some incidents, most of them due to the lack of knowledge that cause some deviations from the initial plan.

First, the choose of DC-DC controllers took more time than the initially expected, in specific, one month more. The reason behind this fact was the introduction of new systems which will need more power for a correct functioning. That's why, at the end, it was decided to select more than one DC-DC controller in order to test them all and pick the one with higher performance or just the one that fits in the requirements needed. What's more, this fact caused that more than one Rear ECU PCB had to be designed in case some of the DC-DC controller does not fulfil the specifications for the applications.

Secondly, the implementation of the wheels speed sensor (phonic wheels) caused problems in the designing of the Rear ECU. The lack of sponsors in this field and the lack of investigation in this sector result in an uncertainty in this method. As a result, the hardware designing for this part was delayed for about one month.

Finally, COVID-19 outbreak led to the cancellation of FSG and FSC competitions, which leaves the team without any kind feedback and comparison with other top international teams. Apart from that, the university and its facilities remained closed from March until June due to the restraints proposed by the government to reduce the number of infected people. Because of this, it was impossible to access to the laboratories and finish the project.

2. State of the art of the technology used or applied in this thesis:

When people think in autonomous cars, they automatically related with a distant future because of the high knowledge and technology required to manufacture this kind of vehicles. But, the thing is that this type of technology exists nowadays and popular automotive brands like Tesla, General Motors or WAYMO have developed a reliable and competent autonomous car prototype. The problem is not in technology but in legal legislation, infrastructure, ethical issues, insurance businesses and much, much more.

Throughout this chapter, the latest investigation and commercial prototypes will be described to give an overall idea of the state of the technology nowadays. Apart from that, it is going to make an emphasis in the electronics of a driverless car and the latest progresses in DC-DC regulators, as it is the main subject of this thesis.

2.1. Autonomous Vehicles (AV)

2.1.1. Tesla

[13] In the public eye, EV manufacturer Tesla became an early leading banner-carrier for advanced driver assistance and self-driving technology.

Tesla pushed its Autopilot software update to properly equipped Model S vehicles in October 2015, enabling auto-steering, lane changing, and parking features. Tesla's deployment strategy and messaging were criticized following a series of crashes and its first Autopilot-driven fatality in the summer of 2016.

Since the accident, Tesla and Mobileye severed ties, with the aim of consolidating control over the development of its radar and camera-based system (avoiding costly LIDAR sensors). This makes Tesla autonomous cars unique, being the only automotive manufacturer who does not use the LIDAR's technology. In order to accomplish that, Tesla has invested a heavy amount of resources to the development of a massive neuronal network for the pedestrians, traffic signals and other cars detection with the camera system.

Starting in October 2016, all Tesla vehicles were built with Autopilot Hardware 2, a sensor and computing package the company said would enable "full self-driving" capabilities once its software matured. Users reported poor performance during the initial rollout of Autopilot 2.0 software, although the system has improved with subsequent updates.

In August 2018, Elon Musk, CEO of Tesla, announced the upcoming release of Autopilot Hardware 3. Model X and Model S vehicles were equipped with Autopilot Hardware 3 and they went into production in March 2019. The system features custom chips manufactured by Tesla, replacing the NVidia Drive platform.

Tesla offers 2 tiers of self-driving capability in its vehicles: Autopilot and Full Self-Driving. Full Self-Driving (FSD) costs an additional $5,000 and includes features like Summon and Navigate on Autopilot, which enables lane changes and interchanges.

Despite the title, Tesla's vehicles are only designated autonomy Level 2 by SAE, meaning they are capable of some autonomous manoeuvres but are not considered fully autonomous. Development is currently underway on a next-generation chip that Musk claims will be "3 times better" than the current chip.

Figure 2.1: Tesla Model 3 Autopilot

Musk also claims that the US electric vehicle maker will be able to launch fully autonomous robotaxis by the end of 2020.

2.1.2. WAYMO (Alphabet)

[13] The Google Self-Driving Car Project is one of the most iconic and initial autonomous vehicle programs. The project formally became WAYMO in December 2016. It expanded its testing beyond Mountain View and Austin to Kirkland, Washington, and Phoenix, Arizona.

WAYMO publicly revealed its custom self-driving hardware in February 2017, planning to sell an integrated hardware and software package. It opened signups for the first public tests of its customized Chrysler Pacifica minivans a couple of months later, quickly followed by its Lyft partnership.

In 2018, WAYMO announced the purchase of 62,000 new Chrysler Pacifica minivans to adapt them to be autonomous vehicles, increasing the size of its self-driving

fleet by about a hundred. Later the same year, the company launched a program to provide residents of the Phoenix area with rides to bus stops and train stations using the autonomous fleet.

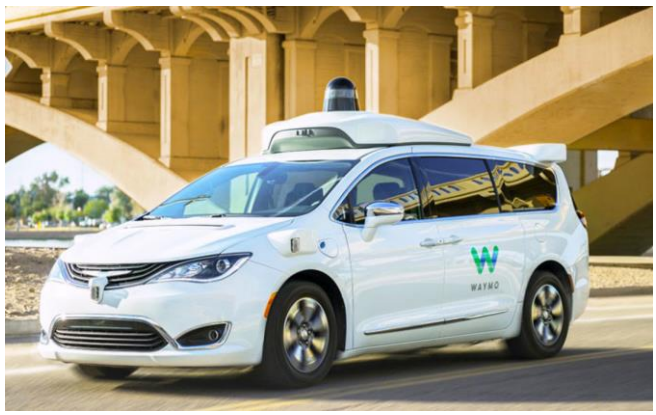


Figure 2.2: WAYMO's van

WAYMO launched WAYMO One, its commercial self-driving ride-hailing service, in Phoenix, Arizona in December 2018. The program is being piloted by a small group of approximately 400 Phoenix residents as part of WAYMO's early rider program.

According to WAYMO, "the self-driving sensor suite consists of LiDAR, cameras, and radar, as well as microphones to detect sounds such as sirens. Like a person's own five senses, WAYMO's self-driving technology is more powerful as a whole than the sum of its parts; each sensor complements the others."

In contrast with Tesla, the main feature of WAYMO's driverless car is its LiDAR installed at the top of the car. WAYMO find indispensable a LiDAR in an AV due to its high reliability and accuracy to detect an object in front at high distances.

2.1.3. ADAS

Advanced driver-assistance systems (ADAS), are electronic systems that help the vehicle driver while driving or during parking. When designed with a safe human-machine interface, they are intended to increase car safety and more generally road safety. ADAS systems use electronic technology such as microcontroller units (MCU), ECUs and power semiconductor devices.

[14] Modern vehicles today have advanced driver-assistance systems integrated to their electronics and manufacturers refresh their car models to add more of these features into their cars. All these systems include adaptive cruise control, park assist, parking assist and traction control.

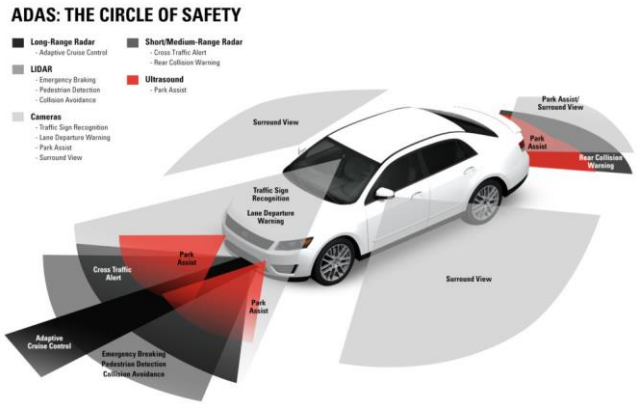


Figure 2.3: ADAS systems

ADAS relies on inputs from multiple data sources, including automotive imaging, LiDAR, radar, image processing, computer vision, and in-car networking.

2.1.4. Smart Cities

Self-driving cars need both vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) technology, and both are still in the early stages of development. [15] The latter requires ubiquitous connectivity—5G, Wi-Fi, DSRC, or some other method—and a multitude of sensors that will need to be installed in streetlights, traffic signals, stop signs, and more. As a result, tones of sensors and IoT devices to modern cities would be needed. The advanced 5G network will allow them to constantly receive and transmit huge amounts of data. Moreover, the steering wheel will be replaced by whatever the driver desires, which will produce more loads on the network.

Automotive vehicles won't just change the way we travel, they will generally change the look of current infrastructure, not only in cities but on highways as well. The AV revolution might bring us to a future of smooth and predictable traffic and more efficient public transportation.

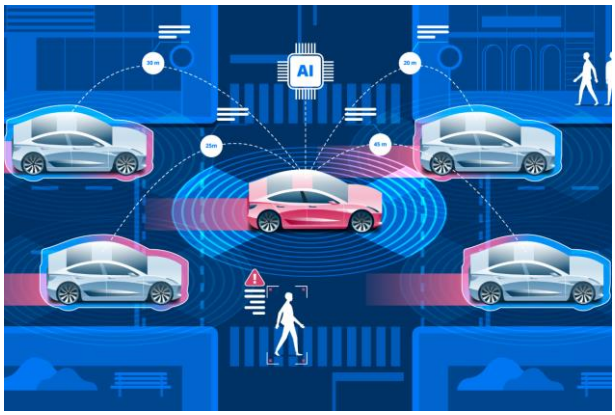


Figure 2.4: Smart Cities connectivity

City residents will have more free space to make use of. Moreover, there will be fewer risks for pedestrians and bicyclists as they have many concerns about urban areas. All of the benefits that autonomous vehicles together with smart cities could bring will improve millions of people's quality of life while taking next-level care for the environment.

2.2. Electronics in AV

Much of the innovation in AV systems centres on optimizing the "virtual driver," or the vehicle's brain. The virtual driver consists of machine-learning algorithms and middleware that connects to the vehicle's sensing, actuation, and communication subsystems. This technology is core to the functioning of the AV.

AVs must have a means to perceive the vehicle's surroundings to determine precise localization, accurately detect and classify fixed and moving objects, and measure the distance to those objects. Most AVs will employ a combination of sensing and perception systems, including camera-based embedded vision systems, radar, and LiDAR sensors. These sensing technologies have complementary strengths, and they can be integrated into an effective sensor suite for AVs.

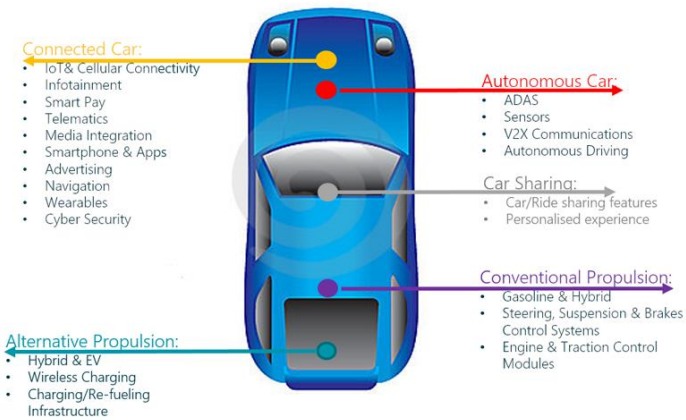


Figure 2.5: Autonomous vehicles diagram

[16] Camera-based vision systems capture rich visual information, but are limited in their ability to determine distance to the object. LiDAR (light detection and ranging) provides precise range measurement with high resolution, but it can be adversely affected by visual obscuration such as fog, smoke, or glare. Conventional radar can "see" through these conditions, but it doesn't produce the high-resolution 3D mapping that can be done with LiDAR. The integration of sensing and perception systems is a focused point by automotive manufacturers due to the usages of cameras, radars and LIDARs for object detection and recognition.

Some embedded vision systems utilize field-programmable gate arrays (FPGAs) and graphics processing units (GPUs), which are well-suited to the high degree of parallelism required by vision-processing algorithms. However, the most successful automotive vision-processing solution to date is the Mobileye EyeQ series, a dedicated hardware-accelerator application-specific integrated circuit (ASIC). An important factor in Mobileye's success in ADAS applications was an extensive period of testing in real-world conditions. This enabled continuous refinement of the algorithms and silicon over successive generations of the chip. It is possible that in the future, AV developers will license their virtual driver software stack to vehicle manufacturers who integrate it onto their platform using standard interfaces for sensors, actuators, and data-communication protocols. However, these standards haven't been fully defined and some sensing technologies aren't mature enough to be separated from the control system.

Apart from all the investment in sensor unit and software, hardware is a focus point for the automotive industry. An example of that are the DC-DC regulators. These devices transform the voltage from the battery to 12V, normally, or the required voltage needed with high efficiency (around the 99%). They have to be significantly robust with high heat dissipation and resistant to the Electromagnetic Interferences (EMIs). Now let's introduce the latest improvements in this field.

2.3. Lately DC-DC Regulators

Practical electronic converters use switching techniques. Switched-mode DC-DC converters convert one DC voltage level to another, which may be higher or lower, by storing the input energy temporarily and then releasing that energy to the output at a different voltage. The storage may be in either magnetic field storage components (inductors, transformers) or electric field storage components (capacitors). This conversion method can increase or decrease voltage. Switching conversion is often more power-efficient (typical efficiency is 75% to 98%) than linear voltage regulation, which dissipates unwanted power as heat. Fast semiconductor device rise and fall times are required for efficiency; however, these fast transitions combine with layout parasitic effects to make circuit design challenging. The higher efficiency of a switched-mode converter reduces the heatsinking needed, and increases battery endurance of portable equipment. Efficiency has improved since the late 1980s due to the use of power FETs, which are able to switch more efficiently with lower switching losses at higher frequencies than power bipolar transistors, and use less complex drive circuitry. Another important improvement in DC-DC converters is replacing the flywheel diode by synchronous

rectification using a power FET, whose "on resistance" is much lower, reducing switching losses.

Most DC-DC converters are designed to move power in only one direction, from dedicated input to output. However, all switching regulator topologies can be made bidirectional and able to move power in either direction by replacing all diodes with independently controlled active rectification. A bidirectional converter is useful, for example, in applications requiring regenerative braking of vehicles, where power is supplied to the wheels while driving, but supplied by the wheels when braking.

[19] Conventional DC-DC converters have virtually no electrical isolation between the input and output circuits; in fact they share a common connection. This is fine for many applications, but it can make these converters quite unsuitable for other applications where the output needs to be completely isolated from the input. These are called Isolating converters and it uses a transformer to accomplish input-output isolation. Another kind of DC-DC regulator is the Charge-pump converter which operates by storing energy as electric charge in a capacitor, instead of using an inductor.

DC-DC converters are available as integrated circuits (ICs) requiring few additional components. Converters are also available as complete hybrid circuit modules, ready for use within an electronic assembly.

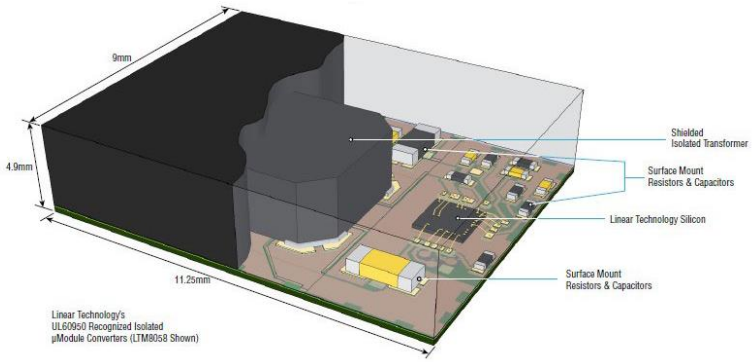


Figure 2.6: DC-DC converter IC

Apart from that the most recent converters and innovations in this field are the modifications of the basic converter which increases the operating regions. The converters can be modified by adding inductor and capacitor which increases the order of the converters. Some examples are adaptive hysteresis control of 3rd order buck converter, predictive controller for fourth order buck converter etc. Multi-input converter can be obtained by adding more than one input sources in the converter.

3. Methodology / project development:

This thesis is divided in two sub-projects and, even though they are related, they are going to be explained separately:

3.1. Rear ECU

The Rear ECU is the ECU placed at the rear of the car and it is in charge of supplying, controlling and acquiring information of the electronic systems placed at the rear.

The first thing to understand the functionality of the Rear ECU was to keep in mind what systems are placed at the rear of the car. For example, the whole Tractive System (TS) of the car (accumulator, motor and inverter) is placed at the rear of the car. Other devices as the TSAL, the brake light, the cooling system and LV battery are also placed at the rear. That's why, the systems and devices previously mentioned are supplied by the Rear ECU.

At the end, it was taken into consideration the previous version of the Rear ECU, the LV ECU, and it was adapted to the new requirements.

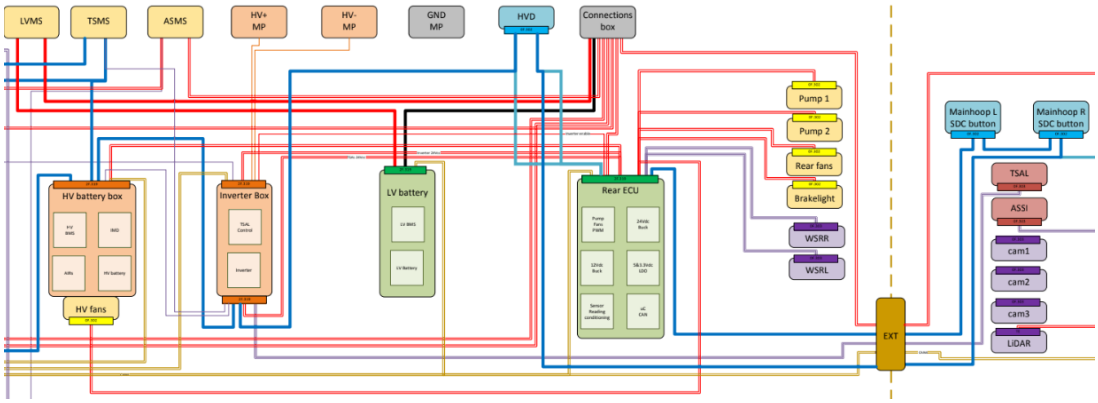


Figure 3.1: Rear ECU Wiring

Studying the new requirements and changes for the new prototype, the Rear ECU would have the following features:

3.1.1. Supply

The Rear ECU is in charge of supplying the devices places at the rear, which are:

- TSAL Controller: PCB installed at the inverter box, which controls and supplies the TSAL.
- Inverter electronics: electronics that controls the inverter functioning.

- Accumulator electronics: electronics that controls the HV battery. Specifically, the Battery Management System (BMS).
- Shutdown Circuit: this circuit has a security purpose and it is defined as a series connection of at least two master switches, three shutdown buttons, the BOTS, the IMD, the inertia switch, the BSPD, all required interlocks and the AMS. If some of these sensors are activated the SDC opens and the car must stop immediately. As the Rear ECU is the nearest to the LV battery, is the one in charge of starting the SDC and supplying the whole sensor systems. The first sensor supplied is the Left Main hoop Button, as it is shown in the Figure 5.1.

Apart from that, the Rear ECU is directly supplied by the LV Battery.

3.1.2. Control

The Rear ECU will have to control key devices placed at the rear:

- Cooling systems: it will have to control cooling system of the inverter and the motor (each one will have its own pumps), the rear fans and the accumulator fans.
- The inverter enable: it will control the enable of the inverter.
- Brake light: it will switch on or off the light.

3.1.3. Sensors

Apart from supplying the whole rear devices, this ECU will have to acquire the information of the following sensors:

- Rear wheels speed sensors: these sensors measure the speed of the rear wheels.
- Temperature sensor: this sensor measures the temperature of key electronic components inside the PCB. It was initially planned to measure the motor temperature but it was found that the motor itself had a temperature sensor integrated within the framework.
- SDC telemetry: the Rear ECU will sense some SDC devices to know if some of them are being activated (for telemetry usages). Specifically, it will sense the High Voltage Disconnect (HVD), the Inverter Interlocks and the Accumulator Container which contains the Insulation Monitoring Device (IMD), the AMS Accumulator Management System (AMS), the main connector and the TSAL sensing.

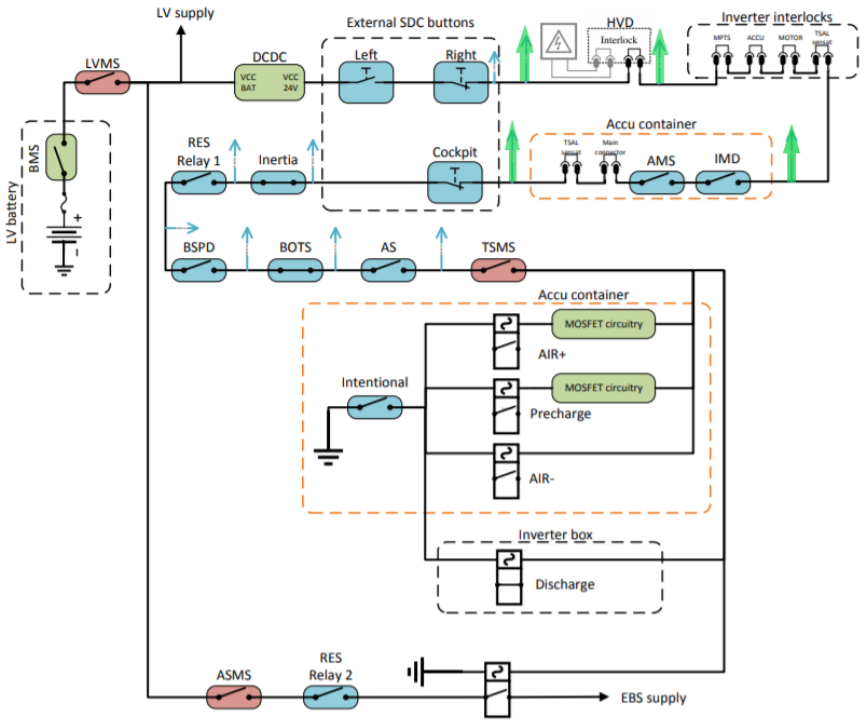


Figure 3.2: SDC with highlighted green flags for Rear ECU sensing

The Rear ECU uses the CAN protocol to communicate and transfer to other modules within the car. Specifically, the Rear ECU receives all the information about the state of the controlled systems and it sends the sensor data to dSpace, which is the brain of the autonomous vehicle.

All the Rear ECU features described could be integrated in the next diagram:

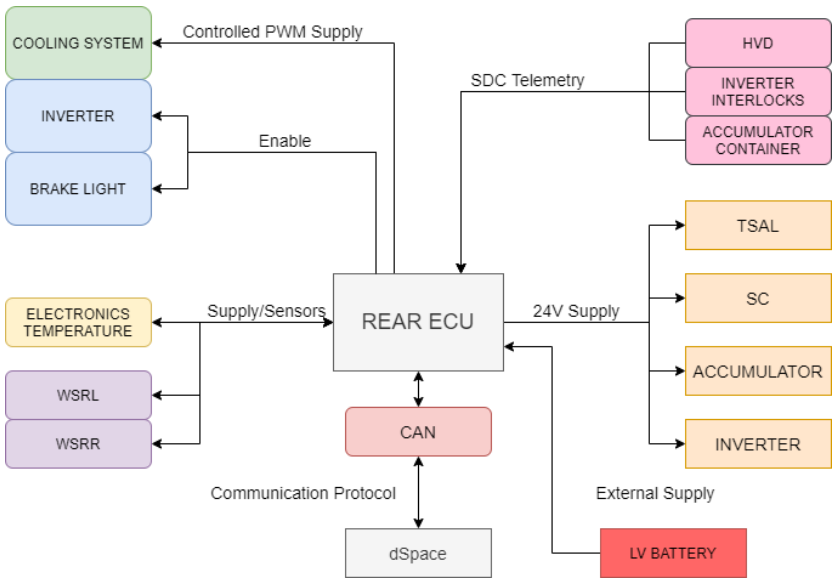


Figure 3.3: Rear ECU Diagram

Having all this requirements and specifications in mind, it was proceeded to design the schematics of the Rear ECU.

3.1.4. Schematics

[2] The schematics of the Rear ECU were designed in the Altium software and they were structured as follows:

- Power stage
- CAN (External communication)
- Microcontroller Unit
- Controlled Outputs
- Sensor unit
- Connectors
- PCB Fan

Here it is only described the most important parts of the Rear ECU schematics. For further information the appendices could be checked.

Before explaining in detail each section, it has to be mentioned that two versions of the Rear ECU were designed with two different 24V DC-DC Controllers in case one of them did not work. That's why, in the power stage are included both designs.

Power stage

The power stage of the Rear ECU is divided in 3 sections:

- 24V: It transforms the voltage from the LV battery (26-30V) to 24V. It supplies the following systems:
  - Shutdown Circuit
  - Inverter electronics
  - TSAL
  - Accumulator electronics
  - Motor cooling system (Pump1)
  - Inverter cooling system (Pump2)
  - Rear cooling system (Fans)
  - Accumulator cooling system (HV_Fans)
  - Brake light
  - Auxiliary 24V (this supply line was designed in case of any system initially non-considered needed 24V. At the end, it was used to supply the wheel speed sensors).

- Other devices within the ECU as the MUX, the current sensors and the PCB fan.

The 24V power stage is formed by a 24V DC-DC regulator which could be broke-down into:

- Commercial DC-DC controller: the first version uses the LM3150 and the second version uses de LM25116. Both are drivers that are able to transform a voltage from 20V to 42V (in this case 30V), to a configurable output voltage (in this case 24V). The LM3150 version is attached in the appendices.
- Transistors: the regulator included two high-quality transistors for the power conversion (BSC059N04LS6ATMA1). They have low on-resistance and low gate charge to minimise the power loses and increase the conversion efficiency.
- Passive components: the rest of passive components (resistors, capacitors and inductors) depends on the specifications each DC-DC controller. The main similarities are the usage of an inductor that could support the maximum current peak.
- LED: a led is installed to have a quick idea if the DC-DC regulator is working.

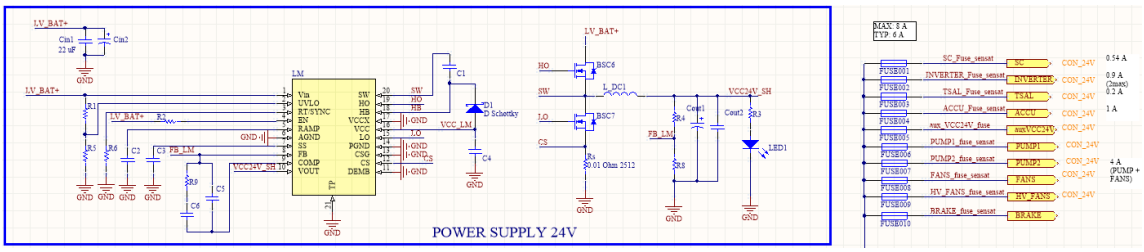


Figure 3.4: 24V power stage Rear ECU

- 5V: It transforms the voltage from the LV battery (26-30V) to 5V. Initially was planned to supply the wheel speed sensors but these needed more voltage for a correct working. Hence, this stage was only used to supply the 3.3V regulator.

The 5V DC-DC regulator composition could be checked in the appendices part.

- 3.3V: It transforms the voltage the 5V regulator to 3.3V. It is used to supply the following devices:
  - Microcontroller
  - CAN controller
  - NTC

The 3.3V DC-DC regulator composition could be checked in the appendices part.

CAN (External communication)

The CAN Unit contains the necessary elements to ensure the CAN communication the communication with other ECUs and systems from the car. It includes the CAN driver and some filtering elements. In the appendices the detailed elements used in this unit are attached.

Microcontroller Unit

The microcontroller unit contains the brain from the ECU, the microcontroller, which receives the information via CAN from external systems and actuates over its systems. The microcontroller used is the STM32F405RGT6 and more detailed information about the components used is shown in the appendices.

Controlled Outputs

The microcontroller unit from the Rear ECU controls the following external devices depending on a CAN message sent by the dSpace:

- Inverter Enable
- Motor Pump
- Inverter Pump
- Rear Fans
- Accumulator fans
- Brake light

To control the mentioned devices, a combination of transistors is used. In this case, two transistors configurations were used to controlled outputs which are explained in the appendices.

Sensor unit

The sensor unit consists of adapting networks for external sensors and some internal sensors installed inside the ECU for telemetry usages. This unit could be breakdown in the wheel speed sensors, the NTC sensors, the current sensors and the multiplexer signals (fuses and SDC signals). Here it is only explained the wheel speed sensors as it is the most important part of the sensor unit. The rest are explained in the appendices section.

- Wheel speed sensor: it consists of an adapting network with a pull-down configuration, which transforms the 24V from the wheel speed sensors to 3.3V

so the microcontroller could read properly the measure. A pull-up configuration was considered in case was necessary. Moreover, a LEMI was installed to reduce noise in the measure although this component could be omitted as the wheel speed sensors a digital, which is more robust in front noise interferences than analogic sensors. A Zener diode is included for protection purposes.

3.1.5. PCB Layout

[2] The Altium software was use for the PCB layout designing.

Rules & restrictions

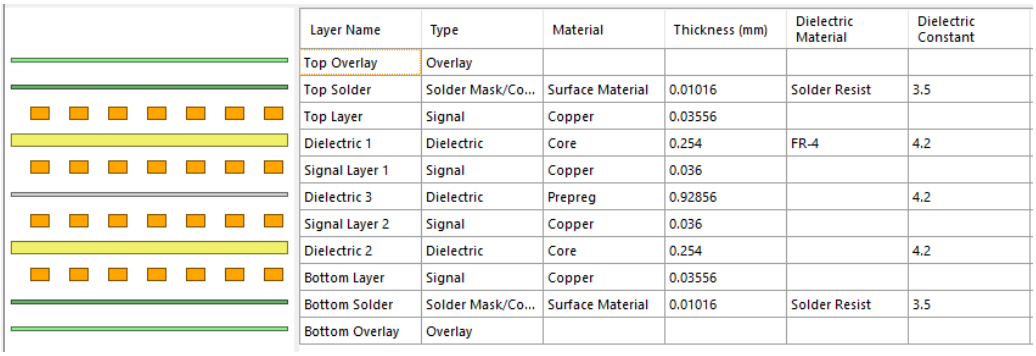
The first thing to design the PCB layout was to establish the rules in Altium software for the layout designing in function of the manufacturer preferences. The rules considered are attached in the appendices.

The external environment where the PCB had to be installed also influenced the PCB design and it introduced some extra limitation to keep in mind in order to devise the PCB layout. For instance, connectors should be installed in the same layer so it would be easier for the wiring design. As a consequence, all high components had to be located to the opposite layer from the layer where the connectors were installed so the PCB could fit well in the box designed.

Following this criteria, the top layer was used for connectors and low height components (as the microcontroller, the multiplexer, SMD components, etc.). The bottom layer was used for high components as the inductor, the 5V regulator or electrolytic capacitors. The interior layers (signal layer 1 and signal layer 2) were used for supply and GND, so both of them could be accessible by vias installation. In fact the supply layer (signal layer 1) was divided in 24V to increase the heat dissipation of the transistor in charge of the 24V convention, and 3.3V to supply the microcontroller and other devices. The GND layer was not divided and the whole layer has the mission of make GND accessible from any point of the PCB.

4 layers

The next step was to define a PCB with four layers and its thickness:



Layer Name	Type	Material	Thickness (mm)	Dielectric Material	Dielectric Constant
Top Overlay	Overlay				
Top Solder	Solder Mask/Co...	Surface Material	0.01016	Solder Resist	3.5
Top Layer	Signal	Copper	0.03556		
Dielectric 1	Dielectric	Core	0.254	FR-4	4.2
Signal Layer 1	Signal	Copper	0.036		
Dielectric 3	Dielectric	Prepreg	0.92856		4.2
Signal Layer 2	Signal	Copper	0.036		
Dielectric 2	Dielectric	Core	0.254		4.2
Bottom Layer	Signal	Copper	0.03556		
Bottom Solder	Solder Mask/Co...	Surface Material	0.01016	Solder Resist	3.5
Bottom Overlay	Overlay				

Figure 3.5: PCB 4 layers characteristics

The exterior layers (top and bottom) were established a 0.03556mm thickness and the interior ones were established a 0.036mm thickness. Between each layer a dielectric was placed to ensure the electrical isolation between layers. Moreover, on top of the top and bottom layer, a solder mask was included to protect the copper of the exterior layers and to add some isolation between the PCB and the exterior environment.

Design process

After establishing the rules and the internal characteristics of the PCB, it was time to design the PCB.

The first thing to do was to establish the measures of the PCB, 55mmx108mm, which were significantly reduced compared to the last year PCB in order to gain space in the rear part of the car and have a more manageable PCB. The PCB need a space to locate the PCB fan so a 40mmx20mm was to need reserved for this purpose. That's why, the right part of the PCB has this squared hole.

After that, the connectors were installed in the top layer on the bottom of the PCB to make the most of the PCB space. From left to right, the first connector is the HEN.2F.308 (power and control), then the HEN.2F.319 (supply, SDC sensing signals, control), next the HEN.2F.312 (CAN, wheels speed sensors, brake light) and finally the HEN.0F.305 (programmer).

The idea was to divide the PCB in two zones, the power zone where all power (24V and 5V regulators) and controlled systems (transistors) would be located, and the microcontroller zone where all sensor signals, multiplexer, microcontroller and CAN would be located. This was done in order to reduce the possible interferences and noise that could cause the inductor and the transistors switching to the control devices (microcontroller and CAN above all). As a result, the left zone was reserved for the power stage and the right zone for the microcontroller and CAN. This fact justify the LEMO disposition mentioned beforehand, which placed two connectors with power lines on the

left zone and the other two connectors with signals, CAN lines and the programmer for the microcontroller on the right side.

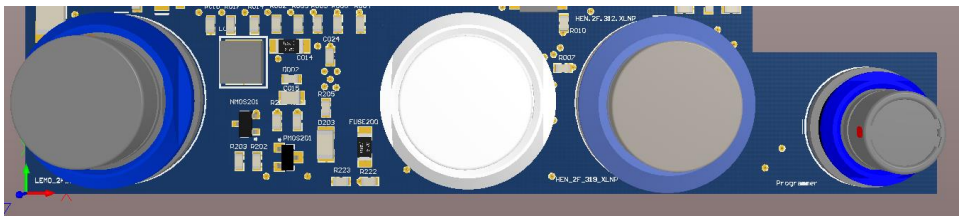


Figure 3.6: Connectors disposition

The most important for the PCB designing were the power zone and the microcontroller and CAN components, which are going to be explained next. The detailed explanation of the rest of the PCB layout is attached in the appendices section. For further information make sure to check this part.

Following the mentioned idea, the power zone was designed. Firstly, the 24V regulator was located, keeping in mind the restrictions of locating the high components as the inductor and the electrolytic capacitors at the bottom layer. Apart from that, a vast amount of space was reserved for the heat dissipation of the transistors and the DC-DC controller. Once it was all components of the 24V regulator located, the 5V regulator was placed near to it at the bottom layer, due to its dimensions. Additionally the current sensors were positioned.

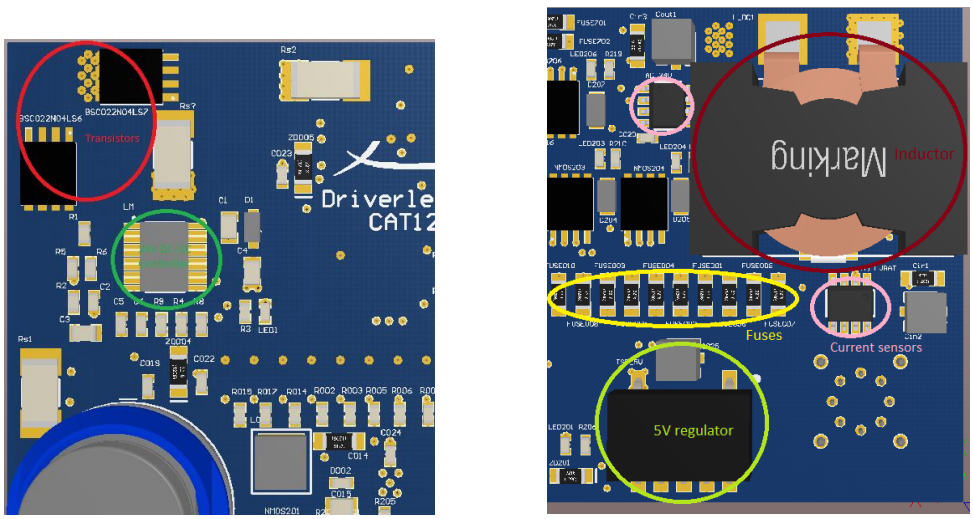


Figure 3.7: PCB Power top & bottom layer

One thing to emphasis was the location of the fuses for the 24V supply systems, which were located in an accessible zone, so if one of them had to be replaced it could be done much easier.

Next, the 3.3V regulator was added. With that, all the components of the microcontroller unit were installed: clock, debugging LEDs, etc. Finally, the CAN unit was designed in the PCB. The components involved in the CAN unit (CAN driver, filter, passive components and transistor) were included as compact as possible and trying to keep the symmetry between all units. The CAN lines were installed as near as possible and with the same length. All this was done with the aim of reducing the external interferences and noise due to the CAN differential protocol.

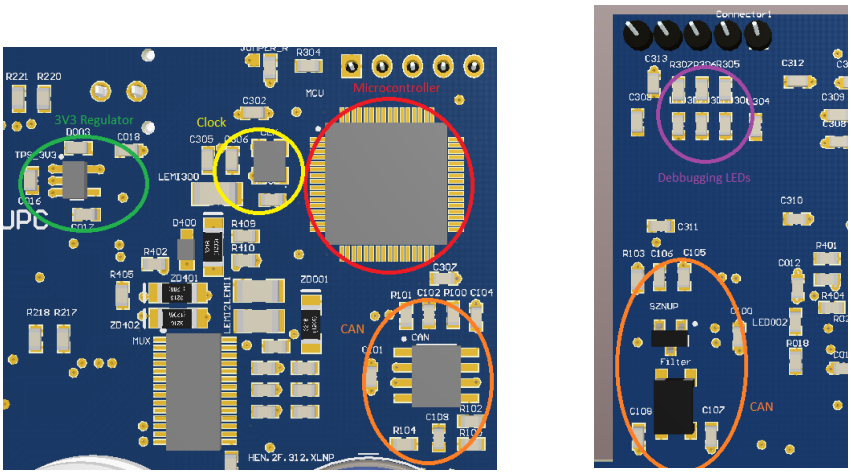


Figure 3.8: Microcontroller Top & Bottom PCB

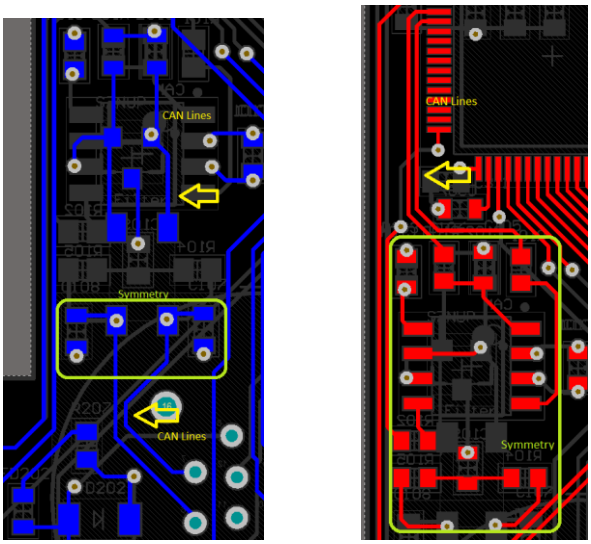


Figure 3.9: CAN lines bottom & top PCB

With all that it was obtained the following PCB. For more detailed information about the general concept check the appendices part.

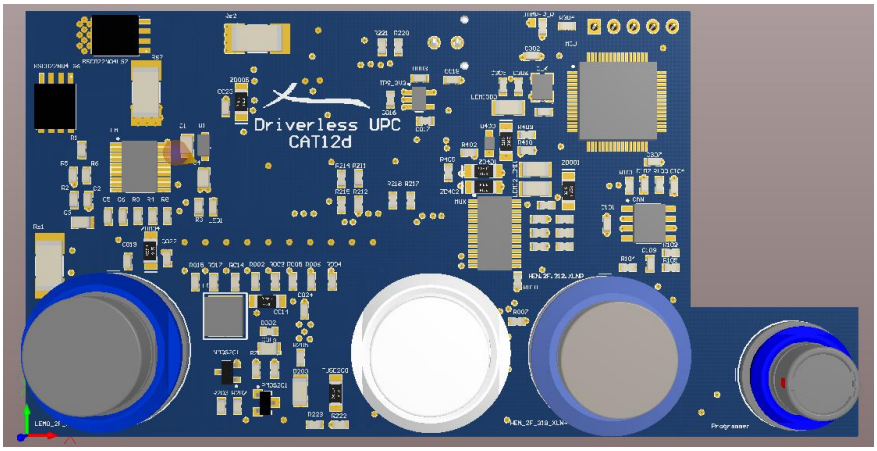


Figure 3.10: PCB Top view

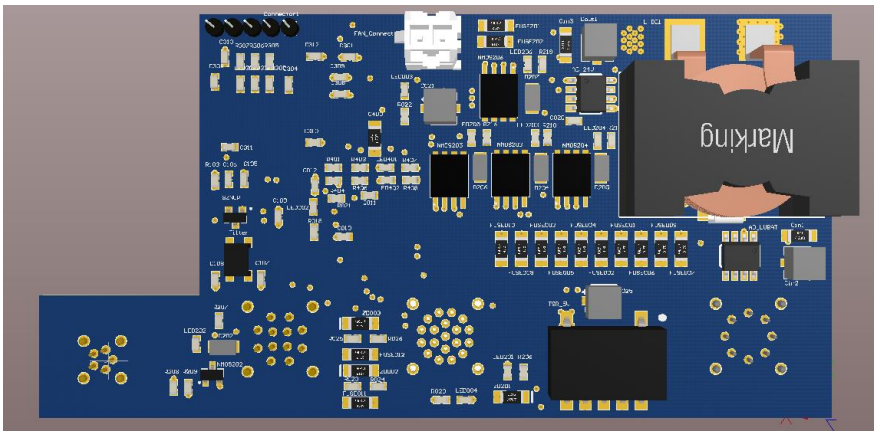


Figure 3.11: PCB Bottom view

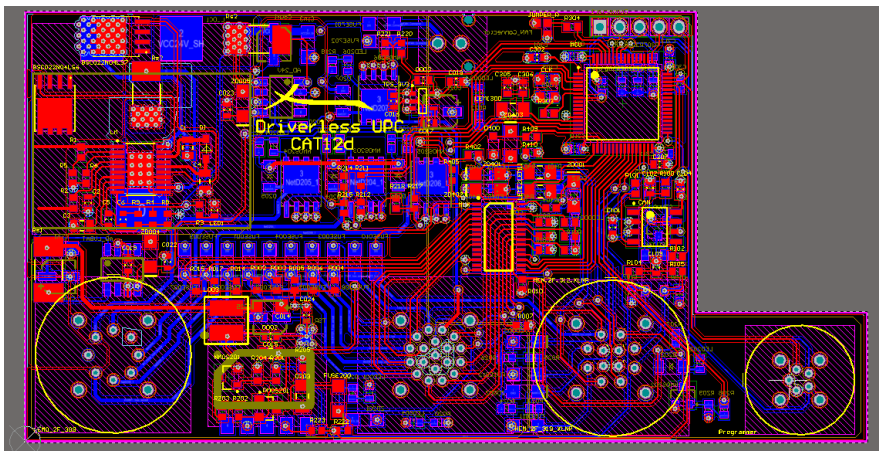


Figure 3.12: PCB Layers

After finishing the PCB layout design in Altium, it had to be validated in the Eurocircuits software (PCB manufacturer) to check the actual rule compliance. To do that, the PCB gerbers (together with the holes gerbers) are exported and uploaded to the Eurocircuits web. After some adjustments the PCB was ready for sending it to manufacture.

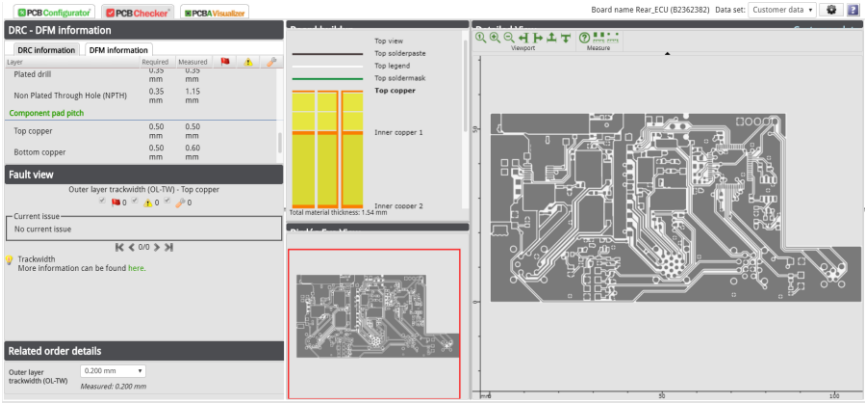


Figure 3.13: PCB validation

3.1.6. PCB Soldering

The PCB soldering was carried out in the UPC installations. A soldering station was used to solder the components to the PCB. High quality tin and flux were involved to make easier the soldering. The procedure followed to solder all components was the next one: it was started with the smaller components, resistors, capacitors and SMD 0603 components; next, it was soldered bigger SMD components, the microcontroller and other IC; and, finally, the big components as the inductor or the electrolytic capacitors were installed, so they interfered the minimum in the soldering procedure.

3.1.7. CAN messages

Once the PCB layout was finished and before the microcontroller programming, the CAN messages were be established so the CAN Database could be developed. The software used to break down the database from all the Can messages implied was the Kvaser Database Editor.

Firstly, it was broken down the information that the Rear ECU must provide to the rest of the car. The most important one was the data acquired from the wheel speed sensor which have to be sent to the dSpace to know the current velocity of the car. This information is crucial so it is used for the control of the car. For this reason, a specific message was created for this purpose. Furthermore, it was established that the data must be provided in 16 bits to get the maximum resolution possible and with a periodicity of 10ms.

Apart from that, the Rear ECU also acquires information from other sensors and signals, which is not essential, but it is important to know the state of part of the car. Hence, this telemetry information is sent to the dSpace each second and it contains the signals from the shutdown circuit, the Rear ECU fuses state, the current sensors data and the PCB transistors temperature.

REAR ECU			
TYPE	INFORMATION	SIZE (Bits)	GROUP
NTC sensing	NTC	8	1
Current Sense	Low Voltage Battery Current	8	1
	24V Current	8	1
Supply fuses and SC sense	BRAKE LIGHT fuse	1	1
	REAR FANS fuse	1	1
	Between Connectors SC	1	1
	TSMS SC	1	1
	HV IN SC	1	1
	HV OUT SC	1	1
	HVD OUT SC	1	1
	HVD IN SC	1	1
	aux24V fuse	1	1
	ACCU fuse	1	1
	TSAL fuse	1	1
	INVERTER fuse	1	1
	SC fuse	1	1
	PUMP1 fuse	1	1
	INV_EN fuse	1	1
WSR sensing	WSRL	16	2
	WSRR	16	2

MESSAGE TO SEND	
REARECU_STATUS	Each 1s

MESSAGE TO SEND	
REARECU_DYNAMICS	Each 10ms

Figure 3.14: Messages to send

The Rear ECU needs the information about the control of the cooling system (rear fans, accumulator fans, inverter pump and motor pump), the brake light and the inverter enable. For this reason, a specific message was created which contains the control variables for the Rear ECU. This message is sent by the dSpace.

TYPE	INFORMATION	SIZE (Bits)	GROUP
Brakelight	BRAKE LIGHT	1	1
Inverter enable	INV EN	1	1
Fans activation	HV FANS	1	1
	REAR FANS	1	1
Pumps activation	PUMP INV (PUMP1)	1	1
	PUMP MOT (PUMP2)	1	1

MESSAGE TO RECEIVE
REARECU_CONTROL

Figure 3.15: Messages to receive

3.1.8. Microcontroller Programming

A microcontroller is a small computer; it contains a CPU along with memory and input/output peripherals. These devices have the advantages that are relatively cheap, small and easy to program and that's why they are used all kind of embedded applications.

[8] The microcontroller used in the Rear ECU is the STM32F405RGTx. [9]This microcontroller contains ADCs (Analogic-to-Digital Converters), CAN bus, timers, DMAs (Direct Memory Access) and interrupts as the most relevant aspects for this application. It also has two APB (Advanced Microcontroller Bus) with different speeds which allows frequencies in timers. Apart from the CAN bus, it includes other communications buses as the USART or de SPI which are not used in this application. Furthermore, for the programming and debugging of the microcontroller, it allows the Serial Wire Debugging (SWD) which is going to be used in this case. Following what it was explained and adding

more details, the microcontroller STM32F405RGTx internal structure is attached in appendices section.

The microcontroller programming was carried out using CubeMX (Microcontroller configuration) and Keil software (Program code). In this section, the program is described and explained.

3.1.8.1. Configuration (CubeMX)

[12]Before starting with the code programming of the microcontroller, the configuration of the microcontroller had to be set to accomplish the requirements and specifications of the Rear ECU. Therefore, the CubeMX software was used to configure the clock, the communication bus, the input/output peripherals, and other necessary applications of the microcontroller. Here, they are explained the most important ones and the rest could be read in the appendices.

CAN

First, the clock configured: it was added an external Crystal/Ceramic Resonator with a 20MHz frequency and it was set a 96MHz internal clock. With that, the software automatically set the internal PLLs (Phase-Locked Loop) to achieve the desired frequency and it allows to observe the frequency used in each system clock (APB1 peripheral, APB1 timer clocks, APB2 peripheral, APB2 timer clocks).

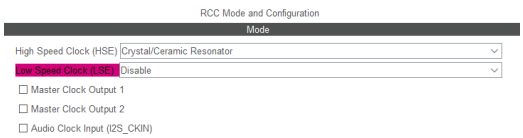


Figure 3.16: Clock configuration: external clock

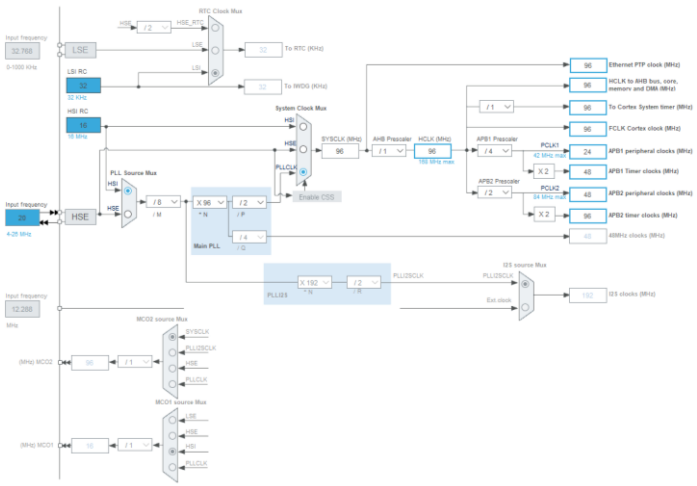


Figure 3.17: Clock configuration: internal clock

Once the clock was configured it was time for the communication bus setting (in this case, the CAN bus). First, the CAN1 was used because it involves the CAN master mode (throughout some experiments the CAN2 -which uses the Slave mode- was causing problems). Then the master mode was enabled and the bit timing parameters were set in order to get a 1Mbit/s speed for the CAN bus.

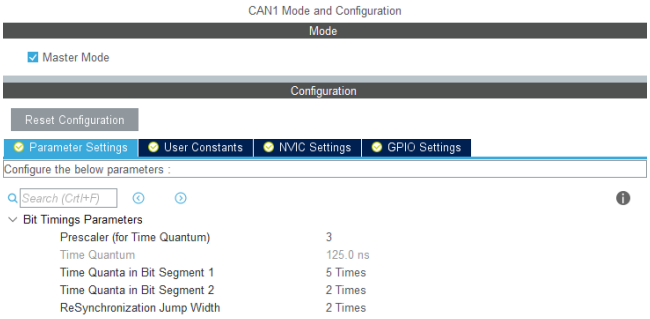


Figure 3.18: CAN Configuration: Bit Timing

The prescaler is set to 3 so a Time Quantum of 125 ns is achieved:

$$Time\ Quantum = \frac{Prescaler \cdot 0.83}{f_{clock}} = \frac{3 \cdot 0.83}{20MHz} = 125\ ns$$

[10] After that, taking into consideration the nominal bit time structure, the Time Quanta in Bit Segment 1 was set to 5 and in Bit Segment 2, it was set to 2. With that we accomplish a baud rate of 1Mbit/s and a sampling point located in the 75% of a bit.

$$Baud\ rate = \frac{1}{(1 + t_{BS1} + t_{BS2}) \cdot Time\ Quantum} = \frac{1}{(1 + 2 + 5) \cdot 125\ ns} = 1Mbit/s$$

$$sampling\ point\ (\%) = \frac{1 + t_{BS1}}{1 + t_{BS1} + t_{BS2}} \cdot 100 = \frac{1 + 5}{1 + 2 + 5} \cdot 100 = 75\%$$

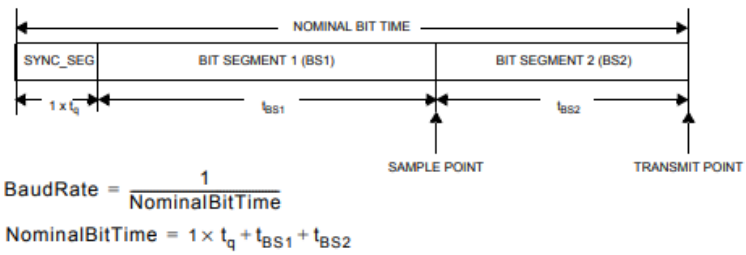


Figure 3.19: CAN STM32 Quantum Bit Time

Additionally, it is enabled the CAN1 RX0 interruption which creates an interruption in the code when a message is received and saved in the FIFO0 of CAN1.

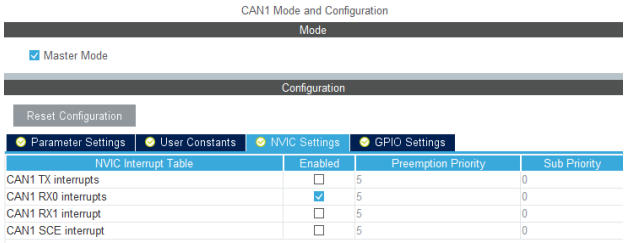


Figure 3.20: CAN Configuration: Interruption

Finally, a GPIO output is defined as CAN enable to activate the CAN controller and the bus functioning.

Wheel speed sensors

The Rear ECU obtains the rear wheels speed by the wheel speed sensors installed in the rear wheels. The model used is the SIEN-M8B-NO-K-L, which is a proximity sensor. This sensor is installed in a cogwheel and it activates a light (signal) when a cog is detected. As a result, an electric pulse is generated, so with its width the speed of the wheel could be calculated.

[11] In order to acquire the information from the wheel speed sensors a couple of timers are needed with the Input Capture direct mode. The Input Capture direct mode generates an interrupt in the timer when a polarity change is detected. This can be handled in the code. With this mode it could be calculated the pulse duration and the period between pulses which is what it is used to calculate the wheel speed.

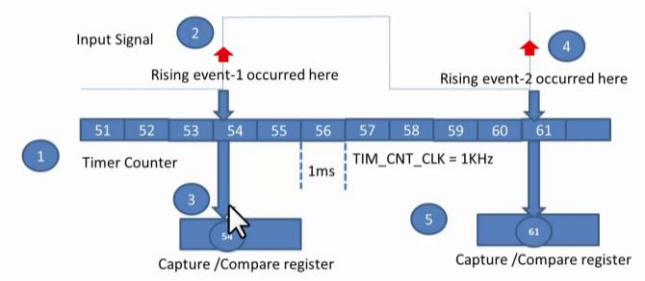


Figure 3.21: Input Capture direct mode

Before starting with the timer's configuration, the formula for the wheel speed had to be found in order to obtain the right sampling frequency and the number of bits of the counter. This was the process it was followed to obtain the wheel speed formula:

$v \equiv wheel\ speed$

$$F \equiv wheel\ frequency = \frac{f}{d \cdot n} = \frac{f}{30 \cdot n}$$

$f \equiv cog\ frequency$

$d \equiv numbers\ of\ cogs = 30$

$n \equiv counter\ number\ (integer: 1,2,3 \dots)$

$R \equiv wheel\ radius = 0.254\ m$

$$v = w \cdot r = 2\pi \cdot F \cdot R$$

$$v = 2\pi \cdot \frac{f}{d \cdot n} \cdot R = 2\pi \cdot \frac{f}{30 \cdot n} \cdot 0.254$$

Once the formula was obtained, it was revealed that the resolution is not constant through all the measures as the counter number is an integer and it is in the divisor. This fact complicated the choice of the key parameters. As a result, it was decided that the measures should have a resolution of at least 0.1 in speeds around 10 m/s. Therefore, a simulation was performed and after adjusting the parameters, it was decided to use a sampling frequency of 100 kHz and 16 bits of information.

```
%%PHONIC WHEELS SIMULATION

bit = 16;
d = 30;
R = 0.254;
f = 100000;

N = 2^bit-1;

v = zeros(1, N);
kmh = zeros(1,N);
n = linspace(1,N, N);

for i = 1:N
    v(i)= f*2*pi*R/(d*i);
    kmh(i) = v(i)*3.6;
end

plot(n,v, '-xr')
title('Phonic Wheels Simulation')
xlabel('Count')
ylabel('V(m/s)')

figure
plot(n,kmh, '-xr')
title('Phonic Wheels Simulation')
xlabel('Count')
ylabel('V(km/h)')
```

Figure 3.22: Wheel speed simulation code

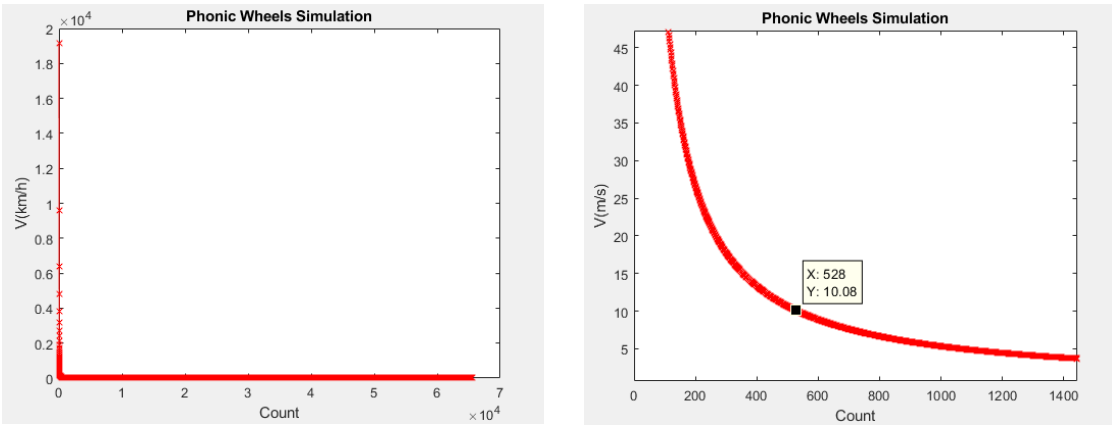


Figure 3.23: Wheel speed simulation plots

First of all, it was chosen a timer in the APB1 bus because it is not needed a high frequency for this application (the APB1 uses 48MHz for timers). As it can be observed the timer 2 and the timer 3 were chosen to carry out this implementation.

Regarding the timer configuration, the Input Capture direct mode is enabled in the channel where the sensor is located. Next, the prescaler is set to obtain the desired frequency, 100 kHz. The counter period is set related with the information bits needed.

$$Prescaler = \frac{f_{clock}}{f_{desired}} - 1 = \frac{48\ MHz}{100\ kKz} - 1 = 479$$

$$Counter\ Period = \ 2^{number\ of\ bits} - 1 = 2^{16} - 1 = 65535$$

The polarity selection is set to rising edge to detect the beginning of the pulse so the periodicity could be calculated.

After that, the timer global interrupt is enabled so it could be handled in the code.

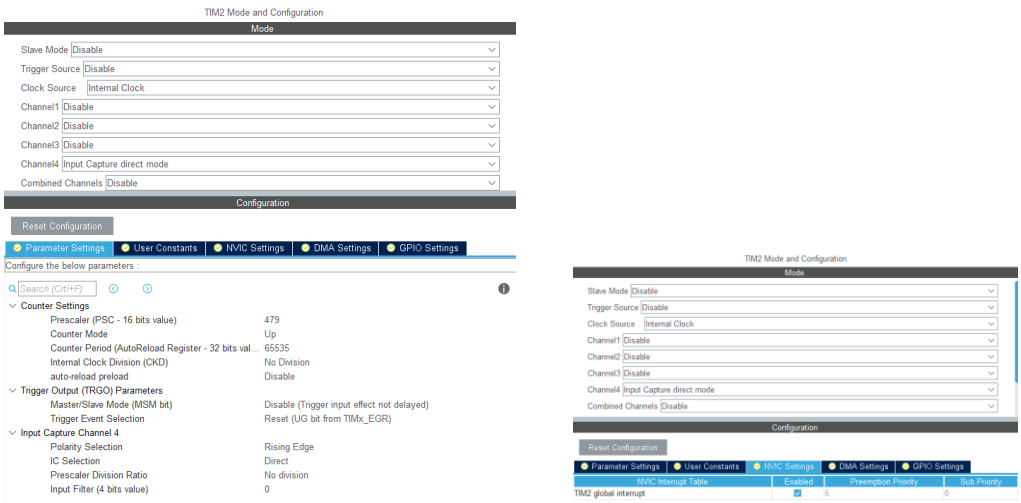


Figure 3.24: Input Capture direct mode: Configuration & Interruption

Due to a mistake in the PCB design, one of the wheel speed sensors signal was installed in a pin without any kind of timer, so this implementation was not possible to accomplish. To fix this, a timer was initialised with the same settings than the one with the Input Capture configuration. To simulate the Input Capture mode, an external interrupt was enabled in that pin, so when a raise edge was detected, an interrupt was handled by the code and could get the counter value from the timer created.

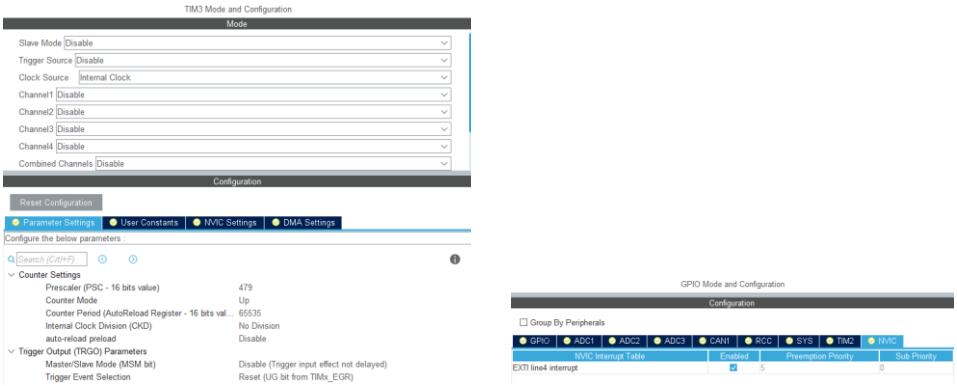


Figure 3.25: External interrupts

Middleware

To accomplish the Rear ECU functionalities a middleware was implemented in the microcontroller programming, in particular, FreeRTOS. This enables the usage of functions (tasks) that are executed periodically in the OS (operative system).

To achieve the desired functionalities for the Rear ECU, 4 tasks were created. All of them have the same priority to avoid any kind of task interruption by an other task with higgher priority.

- SendCANHigh: this task is in charge of sending the dyanamics (rear wheel speed) message. It is sent each 10 ms.
- SendCANLow: this task is in charge of sending the Rear ECU status message. It is sent each 1 s. As a result of the high difference between the periodicty of each message, both send tasks are separated in order to make easier the code understanding.
- ReadCAN: this task is in charge of reading the CAN message received and actuate over the controlled systems.
- ReadMUX: this task is in charge of reading the multiplexer and save the sensor signals in an internal variable.

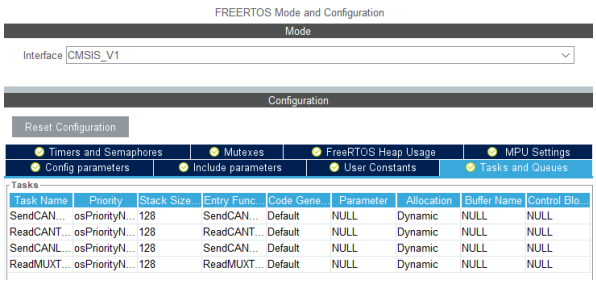


Figure 3.26: FreeRTOS: Tasks

For knowing more about the middleware configuration it is recommended to check the appendices section.

GPIO

Finally, the GPIO are configured as follows:

- Output: HV_FANS. PUMP1, PUMP2, REAR_FANS, LED1, LED2, LED3, MUX_A0, MUX_A1, MUX_A2, MUX_A3, CAN_EN, BRAKE_LIGHT, INV_ENABLE.
- Input: MUX_OUT, INV_EN_FUSE.

The Figure 9.19 shows the entire microcontroller configuration.

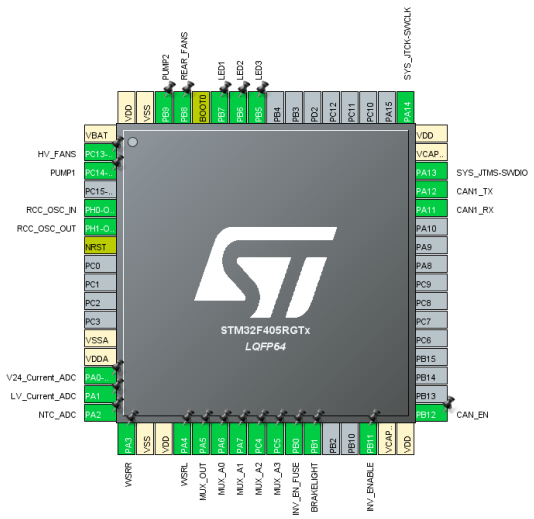


Figure 3.27: Microcontroller configuration

3.1.8.2. Code Programming (Keil)

The entire configuration done in the CubeMX software is initialised in the code. GPIOs, clock, CAN, ADCs, timers and the middleware implemented are included in the code automatically with the setup configured. For example, in the next two figures it can be observed the GPIO and the FreeRTOS initialization:

```
#define HV_FANS_Pin GPIO_PIN_13
#define HV_FANS_GPIO_Port GPIOC
#define PUMP1_Pin GPIO_PIN_14
#define PUMP1_GPIO_Port GPIOC
#define V24_Current_ADC_Pin GPIO_PIN_0
#define V24_Current_ADC_GPIO_Port GPIOA
#define LV_Current_ADC_Pin GPIO_PIN_1
#define LV_Current_ADC_GPIO_Port GPIOA
#define NTC_ADC_Pin GPIO_PIN_2
#define NTC_ADC_GPIO_Port GPIOA
#define WSRR_Pin GPIO_PIN_3
#define WSRR_GPIO_Port GPIOA
#define WSRL_Pin GPIO_PIN_4
#define WSRL_GPIO_Port GPIOA
```

```
osThreadDef(SendCANHighTask, SendCANmsgHighTaskFunction, osPriorityNormal, 0, 128);
SendCANHighTaskHandle = osThreadCreate(osThread(SendCANHighTask), NULL);

/* definition and creation of ReadCANTask */
osThreadDef(ReadCANTask, ReadCANTaskFunction, osPriorityNormal, 0, 128);
ReadCANTaskHandle = osThreadCreate(osThread(ReadCANTask), NULL);

/* definition and creation of SendCANLowTask */
osThreadDef(SendCANLowTask, SendCANmsgLowTaskFunction, osPriorityNormal, 0, 128);
SendCANLowTaskHandle = osThreadCreate(osThread(SendCANLowTask), NULL);

/* definition and creation of ReadMUXTask */
osThreadDef(ReadMUXTask, ReadMUXTaskFunction, osPriorityNormal, 0, 128);
ReadMUXTaskHandle = osThreadCreate(osThread(ReadMUXTask), NULL);

/* USER CODE BEGIN RTOS_THREADS */
/* add threads, ... */
/* USER CODE END RTOS_THREADS */

/* Start scheduler */
osKernelStart();
```

Figure 3.28: Keil Initialization

After the microcontroller configuration, it was time to start with the microcontroller code programming in the Keil software.

Before starting with the code development, a series of constants was defined, this involves the delays of each task, the CAN messages headers, etc.:

```
#define DELAY_HIGH_CAN_MS 10
#define FULL_QUEUE_MS 100
#define EMPTY_QUEUE_MS 100
#define CAN_REARECU_CONTROL 1302
#define DELAY_LOW_CAN_MS 1000
#define CAN_REARECU_STATUS 1314
#define CAN_REARECU_DYNAMICS 1301
#define ADC_BUF_LEN 16
#define DELAY_READ_CAN_MS 100
#define CAN_DLC_REARECU_STATUS 5
#define CAN_DLC_REARECU_DYNAMICS 4
#define DELAY_READ_MUX_MS 50
```

Figure 3.29: Constants

The code could be broke down into four parts in relation with its function (the less relevant parts, such the analogic and the multiplexer signals, are included in the appendices):

- CAN
- Wheel Speed sensors
- Analogic signals
- Multiplexer signals

Wheel Speed Sensors

First, the Input Capture direct mode is started in the timer initialization so the pulse emitted by the wheel speed sensor is captured by the microcontroller. Moreover, the interruption generated by the timer when it finishes the counting is enabled in order to restart the counter.

```
/* USER CODE BEGIN TIM2_Init 2 */
HAL_TIM_Base_Start_IT(&htim2);
HAL_TIM_IC_Start_IT(&htim2,TIM_CHANNEL_4);
/* USER CODE END TIM2_Init 2 */
```

Figure 3.30: WS: Timer Interruption enabling

After that, the Input Capture direct mode interruption and the external interruption call-backs are called in the code. These functions are executed when a rising edge is detected in the corresponding input. As the main goal is to measure the pulse period, the counter value is saved in a 16 bits variable which corresponds to the pulse period and the counter is restarted. This information is used in the “sendCANHighTask” function to send it to the dSpace.

```
uint16_t wsrr_speed = 0; //WSRR value
uint16_t wsrl_speed = 0; //WSRL value
```

Figure 3.31: WS: variables

```c
void HAL_TIM_IC_CaptureCallback(TIM_HandleTypeDef *htim){
  if (htim->Instance==TIM2){
    HAL_GPIO_TogglePin(LED1_GPIO_Port, LED1_Pin); //debbug LED
    wsrr_speed=__HAL_TIM_GET_COUNTER(htim); //WSRR CONT
    __HAL_TIM_SetCounter(htim,0);

  }
}
```

```c
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin){
  wsrl_speed=__HAL_TIM_GET_COUNTER(&htim3); //WSRL CONT
  HAL_GPIO_TogglePin(LED2_GPIO_Port, LED2_Pin); //debbug LED
  __HAL_TIM_SetCounter(&htim3,0);
}
```

Figure 3.32: WS: IC Interruption & External Interruption

When the counter gets to the end, it restarts its value.

```c
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
{
  /* USER CODE BEGIN Callback 0 */

  /* USER CODE END Callback 0 */
  if (htim->Instance == TIM14) {
    HAL_IncTick();
  }
  /* USER CODE BEGIN Callback 1 */
  else if (htim->Instance == TIM2) {
    wsrr_speed=__HAL_TIM_GET_COUNTER(htim); //WSRR CONT
    __HAL_TIM_SetCounter(htim,0);
  }else if(htim->Instance == TIM3) {
    wsrl_speed=__HAL_TIM_GET_COUNTER(htim); //WSRL CONT
    __HAL_TIM_SetCounter(htim,0);
  }
  /* USER CODE END Callback 1 */
}
```

Figure 3.33: WS: End counter interruption

CAN

The CAN is divided in the sending protocol and the reading protocol. The sending part is formed by two tasks “SendCANHigh” and “SendCANLow”, each one is in charge of sending a type of message with the information necessary. The reading protocol involves the interruption to capture de message received and the task “ReadCAN” to do the treatment and actuate over the controlled systems in function of the message received. Part of the CAN code is attached within the memory. If the whole code, the appendices could be checked.

The CAN initialization was modified and it was included some specific parameters for this ECU. First, the CAN enable was activated and the CAN functionality of the microcontroller was started. Apart from that, a filter was included to process only the desired CAN messages. In this case, it is only needed to accept one message, the “RearECU_Control”, which contains the state of the controlled systems. For this reason, the mask applied in the filter, 0xFFFF, only accepts one ID (CAN_REARECU_CONTROL). Furthermore, the CAN auto recovery function is enabled, which allows the microcontroller to restart the CAN bus in case of an oversaturation.

```c
/* USER CODE BEGIN CAN1_Init 0 */
HAL_GPIO_WritePin(CAN_EN_GPIO_Port,CAN_EN_Pin,GPIO_PIN_SET);
CAN_FilterTypeDef sFilterConfig;
/* USER CODE END CAN1_Init 0 */
```

```
if(HAL_CAN_Start(&hcan1) != HAL_OK){
  Error_Handler();
}
//We only accept 0x200 ID
sFilterConfig.FilterBank = 0;
sFilterConfig.FilterMode = CAN_FILTERMODE_IDMASK;
sFilterConfig.FilterScale = CAN_FILTERSCALE_32BIT;
sFilterConfig.FilterIdHigh = CAN_REARECU_CONTROL << 5;
sFilterConfig.FilterIdLow = 0x0000; //CAN BE ANY NUMBER
sFilterConfig.FilterMaskIdHigh = 0xFFFF;
sFilterConfig.FilterMaskIdLow = 0x0000;
sFilterConfig.FilterFIFOAssignment = CAN_RX_FIFO0;
sFilterConfig.FilterActivation = ENABLE;
sFilterConfig.SlaveStartFilterBank = 14;

if(HAL_CAN_ConfigFilter(&hcan1, &sFilterConfig) != HAL_OK)
{
/* Filter configuration Error */
Error_Handler();
}

//Activate CAN autorecovery from Bus-Off State
CAN1->MCR = CAN1->MCR | 0x00000040;

HAL_CAN_ActivateNotification(&hcan1, CAN_IT_RX_FIFO0_FULL);
HAL_CAN_ActivateNotification(&hcan1, CAN_IT_RX_FIFO0_MSG_PENDING);
/* USER CODE END CAN1_Init 2 */
```

Figure 3.34: CAN init

As it is mentioned, the sending protocol has two functions:

- SendCANHigh: This task is in charge of sending the “RearECU_Dynamics” message, which contains the rear wheel speed. It acquires the information from the wheel speed sensors by the variable used to save this specific information. It is executed each 10ms to send periodically the message.

```
void SendCANmsgHighTaskFunction(void const * argument)
{

  /* USER CODE BEGIN 5 */
  TickType_t xLastWakeTime;
  xLastWakeTime = xTaskGetTickCount();

  CAN_TxHeaderTypeDef txHeader;
  uint32_t ptxMailbox;

  /* Infinite loop */
  for(;;)
  {
    uint8_t data2send[8] = {0};

    //Set type of the ID
    txHeader.IDE = CAN_ID_STD;
    //Set ID
    txHeader.StdId = CAN_REARECU_DYNAMICS;
    //Set RTR -> Is Data!
    txHeader.RTR = CAN_RTR_DATA;
    //Do not send timeStamp
    txHeader.TransmitGlobalTime = DISABLE;
    //Set length
    txHeader.DLC = CAN_DLC_REARECU_DYNAMICS;

    //Set data bytes
    wsrr = 100000*2*3.14159*0.254/30.0/wsrr_speed; //debugging variable
    wsrl = 100000*2*3.14159*0.254/30.0/wsrl_speed; //debugging variable
    data2send[0] = wsrl_speed;
    data2send[1] = wsrl_speed>>8;
    data2send[2] = wsrr_speed;
    data2send[3] = wsrr_speed>>8;

    //HAL_GPIO_TogglePin(LED1_GPIO_Port, LED1_Pin); //debbug LED

    //Add TxMessage
    HAL_CAN_AddTxMessage(&hcan1,&txHeader,data2send,&ptxMailbox);
    vTaskDelayUntil(&xLastWakeTime,pdMS_TO_TICKS(DELAY_HIGH_CAN_MS));
  }
  /* USER CODE END 5 */
}
```

Figure 3.35: SendCANHighTask code

- SendCANLow: This task is in charge of sending the “RearECU_Status” message, which contains the information from the current and temperature sensors, and the multiplexer. The multiplexer data is acquired by the buffer filled in the multiplexer and task. The “read_ADC” function is called to get the average from the analogic measurements. The inverter enable fuse input is read and added to the message. The task is executed each second to send the message periodically. The code is attached in the appendices.

The reading protocol consists of the CAN filter configuration (which was already explained), the message capture and the message treatment or processing. The message capture and the message treatment are done separately, because when the message is received, the microcontroller generates an interruption to handle this message. The interruption has to be short timewise to not interfere with the task systems (middleware). That's why, in the interruption the message is saved in an internal variable and it notifies to the task "ReadCAN" where the message treatment is performed. In the "ReadCAN" task, the information inside the message is obtained and the microcontroller actuates over the controlled systems. The task code is included in the appendices section.

```
uint8_t RXCAN_REARECU_CONTROL_DATA[8] = {0};
uint8_t RXCAN_REARECU_CONTROL_ID = 0;

void HAL_CAN_RxFifo0MsgPendingCallback(CAN_HandleTypeDef *hcan){
  BaseType_t pxHigherPriorityTaskWoken;
  CAN_RxHeaderTypeDef rxHeader;
  uint8_t rxData[8];

  if (hcan->Instance==CAN1){
    HAL_CAN_GetRxMessage(&hcan1,CAN_RX_FIFO0,&rxHeader,rxData); //GET DATA
    if(rxHeader.StdId == CAN_REARECU_CONTROL){
      RXCAN_REARECU_CONTROL_ID = 1;
      RXCAN_REARECU_CONTROL_DATA[0] = rxData[0];
      vTaskNotifyGiveFromISR(ReadCANTaskHandle, &pxHigherPriorityTaskWoken);
    }
  }
}
```

Figure 3.36: Read CAN interruption code

3.2. DC-DC Regulator

For supplying all the electronic systems of the car, a LV battery was designed and manufactured last year. This battery is made of Lithium batteries and it provides a voltage maximum voltage of 30V. The problem of using the battery voltage directly to supply the electronics is that when the battery starts to discharge, the voltage drops, and, as a consequence, the supply would not be constant. In order to have a continuous supply source, voltage regulators were involved and installed in each ECU. In general, the voltage used for the electronics supply is 24V, to reduce the current demand from the devices, but in some applications 12V were used.

A voltage regulator is an electrical system designed to automatically maintain a constant voltage level from a variable source. In this case, the voltage is transformed from a DC source (LV battery) to a DC supply (12V-24V). For this application, a high efficiency and low heat generation was needed, which made the switching voltage regulator a perfect candidate for the voltage regulation.

A switching regulator is a voltage regulator that uses a switching element (transistors and diodes) to transform the incoming power supply into a pulsed voltage, which is smoothed using capacitors, inductors, and other elements. Power is supplied

from the input to the output by turning ON a switch, charging the inductor, until the desired voltage is reached. Once the output voltage reaches the predetermined value the switch element is turned OFF and no input power is consumed; the inductor discharges and provides current to the load. Hence, the duty cycle of the switch set how much charge is transferred to the load. This is controlled by a difference feedback between the desired voltage and the real output voltage, with the goal of reducing the error between these two. Repeating this operation at high speeds makes it possible to supply voltage efficiently (due to the low power dissipation from the transistors) and with less heat generation.

There are different switching regulators configurations depending on the output voltage needed in relation with the input voltage. In this case, a step-down or Buck converter was used as the output voltage is lower than the input voltage. For this reason, the typical Buck configuration, which is revealed in Figure 10.2, was used.

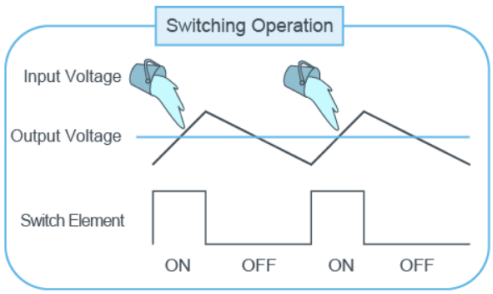


Figure 3.37: Switching Regulator functioning

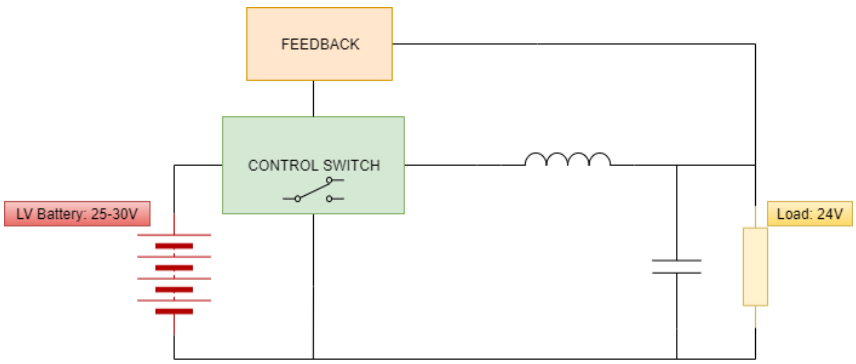


Figure 3.38: Switching Regulator Configuration

Last year, commercial buck regulators were used to transform the LV battery to 12V and 24V for the electronics supply. However, these voltage regulators had considerably low efficiency (around the 70%) and they suffered significant overheating. That's why, the aim for this year was to design a reliable switching regulator with high efficiency and low overheating.

The first idea was to control the switch system with the microcontroller. The problem was that some ECUs had the microcontroller saturated, so not all ECUs could use this system. For this reason, it was decided to use commercial switching voltage controller. This controller gets a feedback signal from the real output voltage, it compares it with the desired output voltage and it monitors duty cycle (ON-OFF switches, transistors) to get the desired output voltage. These drivers provide high efficiency and low overheating which was ideal for this application. Moreover, it allows tuning and choosing the best electronic components for certain applications, which helps to have a less power dissipation and, as a result, higher efficiency.

Each ECU installed in the car needed a voltage regulator to convert the voltage from the LV battery to a stable 24V supply. But, each one had different characteristics, requirements and necessities. This is why, for the voltage controller selection, two main applications were defined depending on the current consumption needed:

- Low current demand: inside this group, it is included the Front ECU and the EBS (Emergency System Brake) Supervisor ECU. Both of these ECUs do not supply high power demanding applications with the 24V. The EBS ECU supplies the EBS and the pressure sensors (2A maximum in total), and the Front ECU supplies the Dash ECU and the Remote Emergency System (RES), (1.5A maximum in total). As a result, simplicity could be prioritized in the design and low complex voltage drivers were searched.
- High current demand: inside this group, it is included the Rear ECU, the PU ECU and Steering PCB. These ECUs supply systems with high power consumption. The Rear ECU supplies all rear systems (7A maximum in total with 24V), the PU ECU supplies the Graphic Processing Unit (GPU), the LiDAR, the buzzers and the Nvidia (6A maximum in total with 12V), and the Steering PCB (10A of peak). Hence, for these applications a more complex controller was needed to accomplish with the applications requirements. Moreover, it was searched some special transistors and inductors for high power applications.

To sum up, the requirements specified for these two applications were the following ones:

- Input voltage range: 24V - 30V
- Output voltage: 24V
- Output current: 2A (low demanding applications) / 10A (high demanding applications)

- High efficiency: 90%

[5] After an exhaustive search, two suitable DC-DC switching controllers were selected: the LM3477, used in the low current demand applications, and the LM25116, used in the high current demand applications.

3.2.1. LM3477

[3] The LM3477 is a high-side N-channel MOSFET step-down (buck) switching regulator controller. Its design is significantly simple as it only uses an N-channel MOSFET and a diode to accomplish the switching control, combined with the feedback resistor network, as it can be observed in the figure 10.3.

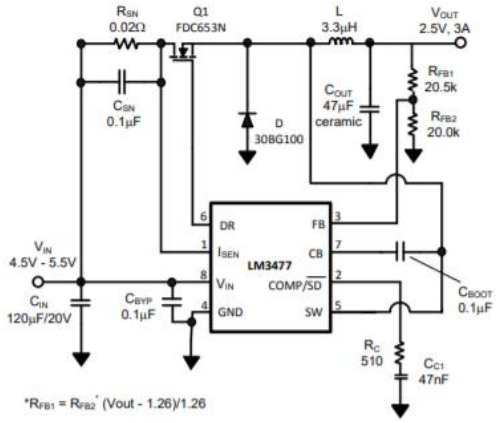


Figure 3.39: LM3477. Typical step-down configuration

Among its features, it stands out a 3V - 30V input range, a 500 kHz fixed switching frequency, a 93% maximum duty cycle, 4A maximum output current, around 90% efficiency. Furthermore, it includes frequency compensation network, an adjustable current limit and an internal soft start.

Its functional diagram it presents low complexity and it is considerably understandable, which makes things easier if something does not work properly. It basically acquires the feedback voltage from the resistor network and compared by an error amplifier with 1.26V. This amplified error is compared to the real output voltage through a PWM comparator, shown in the Figure 10.5. The output from the comparator is used by the Switch Logic block to calculate the duty cycle necessary to achieve the desired output voltage and actuates over the switch driver and, consequently, over the transistor control signal.

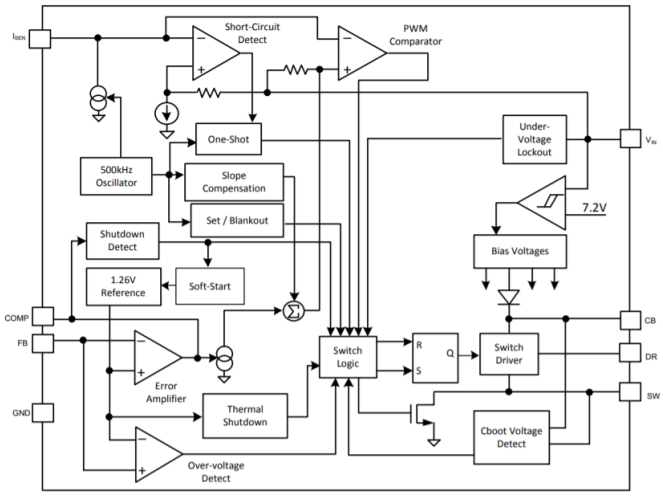


Figure 3.40: LM3477. Functional Block Diagram

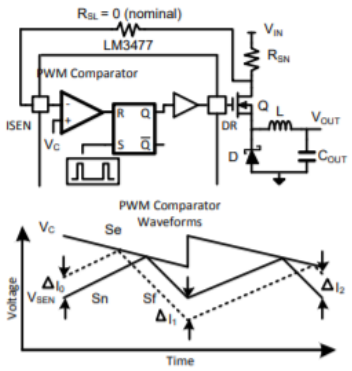


Figure 3.41: LM3477. PWM Comparator and Feedback loop

Components selection

After the controller study, it was time to select the components needed to complete the power schematics of each ECU. As more of one ECU used this voltage controller it will be shown the formulas involved in the components selection and calculation, but not numeric value will be specified.

The duty cycle is the fraction of one period in which the system is active. In buck regulators it could be approximated to the following expression:

$$D \approx \frac{V_{out}}{V_{in}}$$

The inductor value is chose keeping in mind the manufacturer recommendations, the current ripple and the continuous conduction mode of the buck regulator. Apart from that, it must be able to support the maximum output current:

Manufacturer recommendations:

$$\frac{V_{IN}\cdot 1.8\cdot R_{SN}\cdot(\frac{1}{\pi\cdot Q_{MAX}}+D-0.5)}{f_s\cdot(V_{SL}+50\cdot 10^{-6}\cdot R_{SL})}\leq L\leq\frac{V_{IN}\cdot 1.8\cdot R_{SN}\cdot(\frac{1}{\pi\cdot Q_{MIN}}+D-0.5)}{f_s\cdot(V_{SL}+50\cdot 10^{-6}\cdot R_{SL})}$$

$Q_{MAX}=2$ $f_s\equiv switching\ frequency\ (500kHz)$

$Q_{MIN}=0.15$

$R_{SL}=0$

$V_{SL}=83$ mV

Current ripple:

$$L=\frac{V_{OUT}(1-D)}{\%\Delta I_L\cdot I_L\cdot f_s}$$

Continuous conduction mode limit:

$$L>\frac{D(V_{IN}-V_{OUT})R_{LOAD}}{2\cdot f_s\cdot V_{OUT}}$$

The output capacitance is calculated in function of the maximum output ripple desired. A combination of electrolytic capacitors and ceramic capacitors (with low ESR) is used to stabilise the output signal and filter it from noise.

$$C_{OUT(MIN)}=\frac{L(V_{OS(MAX)}-\sqrt{V_{OS(MAX)}^2-\left(\Delta I_{OUT(MAX)}\cdot R_{ESR}\right)^2}}{V_{OUT}\cdot R_{ESR}^2}$$

The transistor implied is selected to have a quick switch with low noise and low power dissipation. That's why the conduction and the switching losses were calculated:

$$P_{cond}=I^2D_{MAX}\cdot r_{DS(ON)}$$

$$P_{sw}=\frac{V_{IN(MIN)}\cdot I_{OUT(MAX)}\cdot Q_{gd}\cdot f_s\cdot(\frac{8.5}{6-V_{th}}+\frac{6.8}{V_{th}})}{2}$$

$$P_{total}=P_{cond}+P_{sw}$$

The power diode is selected to support the average current through it. Moreover, a Schottky diode was installed to reduce the switching noise.

$$I_{D(AVG)}=I_{OUT}\cdot(1-D)$$

The input capacitors smooth the input voltage and filter the input noise. A combination of electrolytic and ceramic capacitors is implemented to accomplish the mentioned fact. The input capacitor could be tuned with any value (values around 1 µF and 200 µF are the most common) The manufacturer recommends installing a 0.1 µF or 1 µF ceramic bypass capacitor very close to pin 8.

The output voltage can be programmed using a resistor divider between the output and the feedback pins as is shown in the Figure 10.3. The resistors are selected such that the voltage at the feedback pin is 1.27V:

$$V_{OUT} = 1.27 * (1 + \frac{R_{FB1}}{R_{FB2}})$$

The LM3477 contains a compensation network to stabilise the control loop and achieve high performance in terms of transient response, audio susceptibility and output impedance. There are several different types of compensation that can be used to improve the frequency response of the control loop. To determine which compensation scheme to use, some information about the power stage is needed.

$$H = \text{feedback gain} = \frac{R_{FB2}}{R_{FB1} + R_{FB2}}$$

$$A_{DC} = \frac{R}{1.8R_{SN}} \frac{1}{1+\frac{R}{f_S L}\left[m_c \times D' - 0.5\right]}$$

$$m_c = 1+\frac{S_e}{S_n}$$

$$S_e = f_S(V_{SL} + 50\times10^{-6}\, R_{SL})$$

$$S_n = \frac{V_{IN}D' \times 1.8R_{SN}}{L}$$

$$f_{p1} = \frac{1}{2\pi}\left[\frac{1}{C_{OUT}R} + \frac{1}{f_s L C_{OUT}}(m_c \times D' - 0.5)\right] (Hz)$$

$$f_{ESR} = \frac{1}{2\pi C_{OUT} R_{ESR}} (Hz)$$

$$Q = \frac{1}{\pi\,(m_c \times D' - 0.5)}$$

Figure 3.42: LM3477. Compensation Power stage equations

With the power stage known, a compensator can be designed to achieve improved performance and stability. First, a target crossover frequency (fc) for the loop gain must be selected. The crossover frequency is the bandwidth of the converter. A higher bandwidth generally corresponds to faster response times and lower overshoots to load transients. However, the bandwidth should not be much higher than 1/10 the switching frequency, so a crossover frequency between 10kHz - 50kHz is recommended.

The strategy taken here for choosing R_C and C_{C1} is to set the crossover frequency with R_C, and set the compensator zero with C_{C1}:

$$R_C = \frac{f_c \cdot R_{GM}}{A_{DC} \cdot GM \cdot R_{GM} \cdot H \cdot f_{p1} - f_C}$$

$f_C \equiv Crossover\ frequency\ (between\ 10kHz\ and\ 50kHz)$

$R_{GM} = 50\ \text{k}\Omega$

$GM = 10^{-3}$ A/V

$$\frac{3.16}{2\pi \cdot f_C \cdot R_C} \leq C_{C1} \leq \frac{1}{2\pi \cdot f_{p1} \cdot R_C}$$

The adjustable current limit is set by the sense resistor R_{SN}. The voltage across R_{SN} is compared to an internal control voltage V_C. The sense resistor must be lower than a maximum value to ensure the correct functioning of the current limit.

$$R_{SN(MAX)} \approx \frac{V_{CL(0)(MIN)} - D_{MAX}(V_{CL(0)(MIN)} - V_{CL(100)(MIN)})}{1.15 I_{OUT(MAX)}}$$

$V_{CL(0)(MIN)} = 125\ \text{mV}$

$V_{CL(100)(MIN)} = 43\ \text{mV}$

Simulation

[6] After that, it was proceeded to simulate the circuit to have an idea of the regulator performance. The tool used to carry out this simulation was the Texas instruments webench (power designer simulator). The EBS ECU was set as an example as it was the most current demanding application. The most relevant components chose for this simulation were a 22 µH, to reduce the current ripple under the 20%; a high quality transistors (BSC059N04LS6ATMA1) with low $r_{ds(on)}$ and Q_{gd}, to reduce power dissipation and increase the efficiency; and 50 µF output capacitance, to decrease the output voltage ripple.

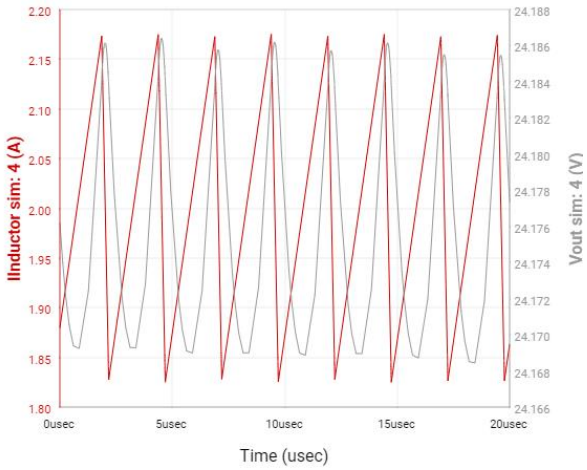


Figure 3.43: LM3477. Output Simulation

As it could be observed in the Figure 10.7, the output voltage stabilise in 24.18V approximately with a low voltage ripple, around the 15 mV, and the output current is 2 A with 0.35A current ripple. Furthermore, the efficiency estimated is around the 95 %.

3.2.2. LM25116

[4] The LM25116 is a synchronous buck controller intended for step-down regulator applications from a high voltage or widely varying input supply. The control method is based upon current mode control using an emulated current ramp. The LM25116 drives external high-side and low-side NMOS power switches. A user-selectable diode emulation mode enables discontinuous mode operation for improved efficiency at light load conditions.

Among its features, it stands out a 6V - 42V input range, a programmable switching frequency from 50 kHz to 1 MHz, a 95% maximum duty cycle, 20A maximum output current, around 99% efficiency. Furthermore, it includes a diode emulation mode (to improve efficiency), a compensation network, a programmable current limit, a programmable under-voltage lockout and a programmable soft start.

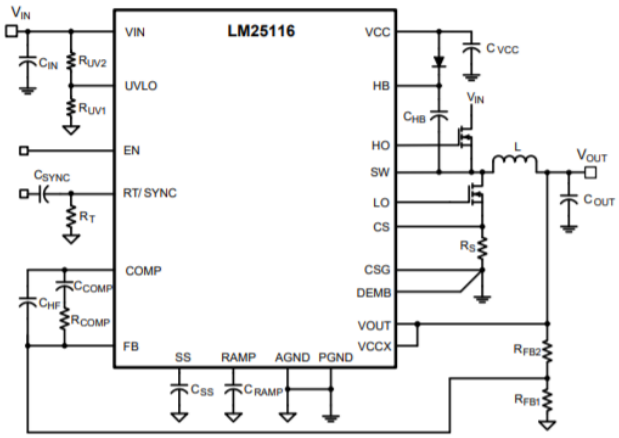


Figure 3.44: LM25116. Typical Application

The functional diagram, Figure 10.10, explains the internal functioning of the controller and it is useful to understand as it could help to solve possible problems. The LM25116, as the LM3477 controller, acquires the feedback voltage from the feedback resistor divisor and compared by an error amplifier with 1.215V. Apart from that, the LM25116 includes a ramp generator that creates a ramp signal in function of the input and output voltage, shown in the Figure 10.9. This ramps signal is compared with the error obtained before through a PWM comparator. The output from the comparator is introduced in the Adaptive Timer block to calculate the duty cycle necessary to achieve the desired output voltage and actuates over the transistors gates (high-side and low-side) to control the voltage regulation.

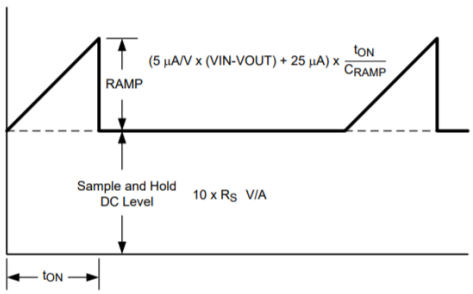


Figure 3.45: LM25116. Ramp Generator

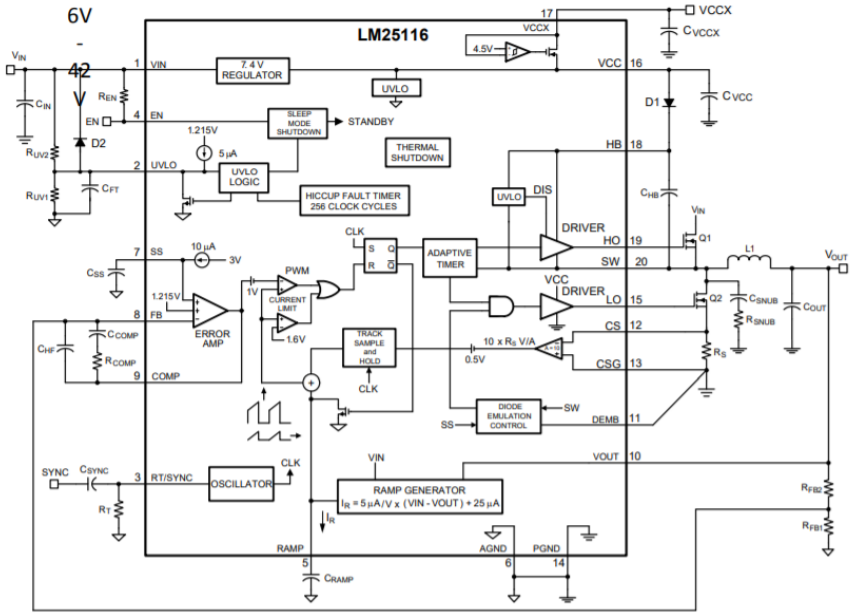


Figure 3.46: LM25116. Functional Block Diagram

Components selection

As it was done with the LM3477, after the controller study, the components were calculated and selected to complete the power schematics of each ECU. As more of one ECU used this voltage controller it will be shown the formulas involved in the components selection and calculation, but not numeric value will be specified. This part is attached in the appendices as it is similar to the LM3477 components selection.

Simulation

[6] After that, it was proceeded to simulate the circuit to have an idea of the regulator performance. The tool used to carry out this simulation was the Texas instruments webench (power designer simulator). The Steering PCB ECU was set as an example as it was the most power demanding application. The most relevant components chose for this simulation were a 4.7 µH, to reduce the current ripple under the 30%; high quality transistors (BSC059N04LS6ATMA1) with low $r_{ds(on)}$ and Q_{gd}, to reduce power dissipation and increase the efficiency; and 100 µF output capacitance with low esr, to decrease the output voltage ripple.

As the LM25116 was destined to high demanding current applications, most of the simulations are focused in the power dissipated in the key devices as the transistors and the IC, and the final efficiency. The efficiency is between 98.5% and 99% as it is shown in Figure 10.11. In the power dissipation charts it can be observed that the high-side NMOS

is the most critical part, as it must be able to support at least 2.5W (due to the high duty cycle). This fact was considered in the PCB layout design and it was reserved some space for this specific transistor for heat dissipation reasons. In the Figure 10.14, the IC power dissipation predicted is way under the limit, 1.5W. Finally, the output voltage and current ripple were simulated to have an idea of the obtained ripple. It could be observed that the voltage ripple is significantly low, around the 0.10V in the worst scenario (Figure 10.15); but the current ripple is considerably high; around the 5A when V_{IN} is 30V (Figure 10.16). As a consequence, an inductor with higher inductances were considered and installed in the real applications.

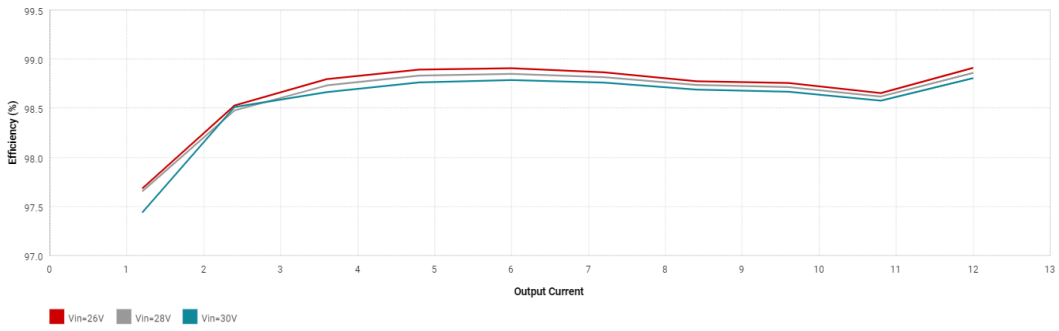


Figure 3.47: LM25116. Efficiency

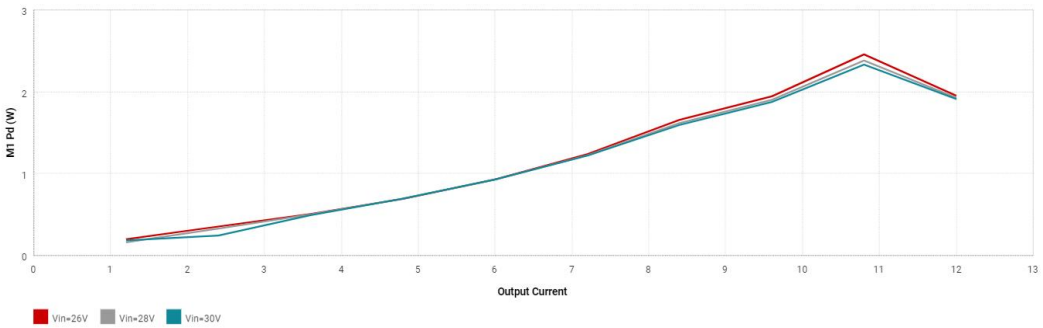


Figure 3.48: LM25116. HO-NMOS power dissipation

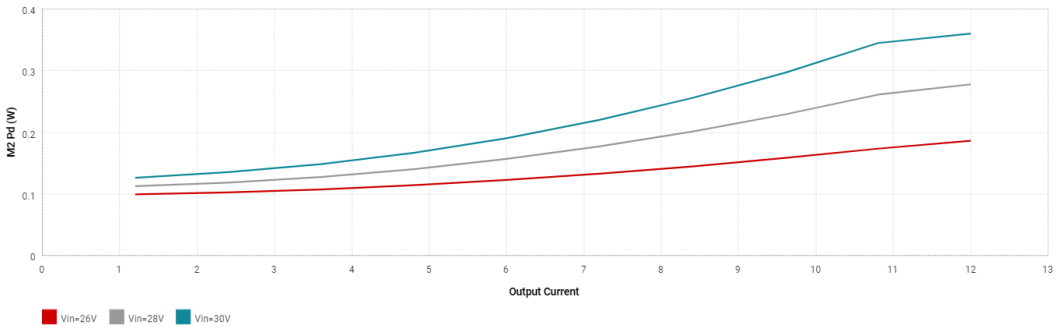


Figure 3.49: LM25116. LO-NMOS power dissipation

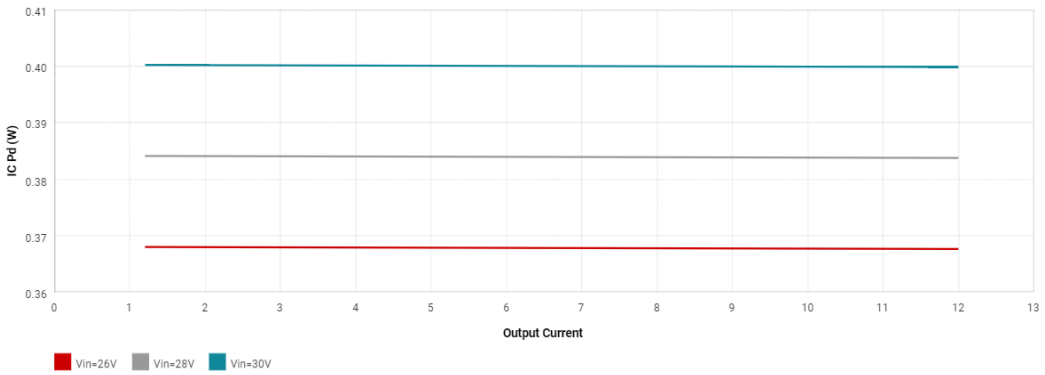


Figure 3.50: LM25116. IC power dissipation

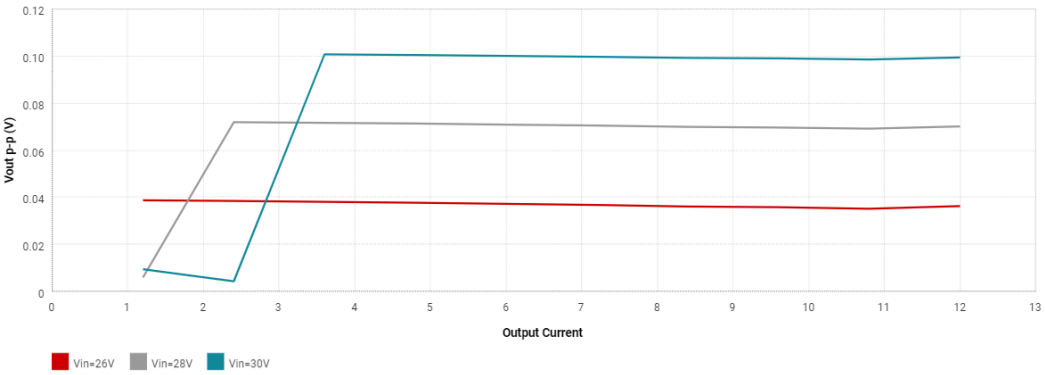


Figure 3.51: LM25116. Output voltage ripple

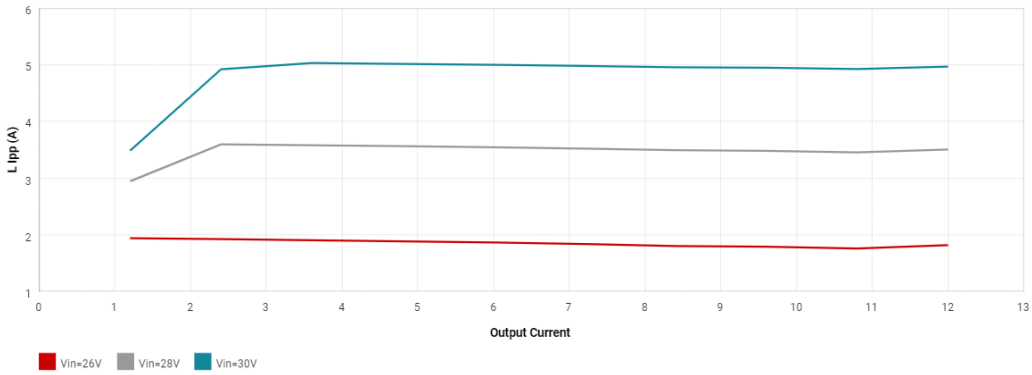


Figure 3.52: LM25116. Output current ripple

4. Results

The results from the project were a functional ECU, the Rear ECU, and high efficiency DC-DC regulators installed in the car.

Firstly, the power stage of the Rear ECU was tested to observe if any failure appeared. After checking the correct supply of the ECU, the Rear ECU was involved in a virtual environment where the CAN communication (sending and receiving CAN messages with different headers and information) with other ECUs was simulated to check its correct functioning. Apart from that, the wheel speed sensors were tested to observe the measures obtained and how they were displayed by the microcontroller. Moreover, other aspects as the read of the Shutdown circuit signals and the ADCs were tested. The KVASER software was used to carry out all these simulations and testing; an example of this communication is shown in the Figure 4.1. With all that, the final functional Rear ECU was obtained.

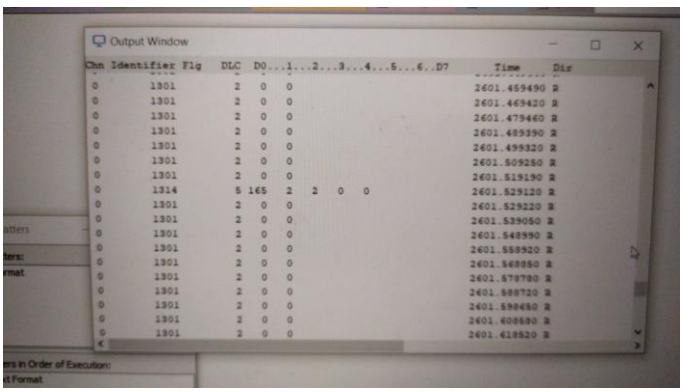


Figure 4.1: CAN simulation in KVASER

Once the final version of the Rear ECU was achieved, it was time to integrate with the other ECUs of the car and to check its functioning in a real environment.

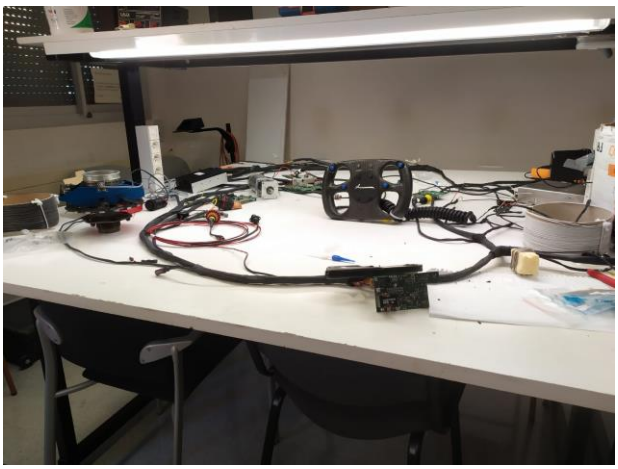


Figure 4.2: Rear ECU connected to the wiring

Finally, once it was integrated, it was planned to install all the ECUs with the wiring inside the car and do a final test of the electronics before letting the car run. But, due to the COVID-19 outbreak, it was not possible to perform the final test in time.

Regarding the DC-DC regulators, first they behaviour was simulated to have an initial idea of their performance (simulations in methodology DC-DC Regulators section). After soldering and installing the DC-DC regulators, they were tested trying to simulate a real environment and reaching the maximum power consumption estimated. Both DC-DC regulators performed satisfactorily with an efficiency of 90-95% (LM3477) and 97-99% (LM25116) as it can be observed in Figure 4.2. These results were considerably similar to the ones obtained in the simulations.

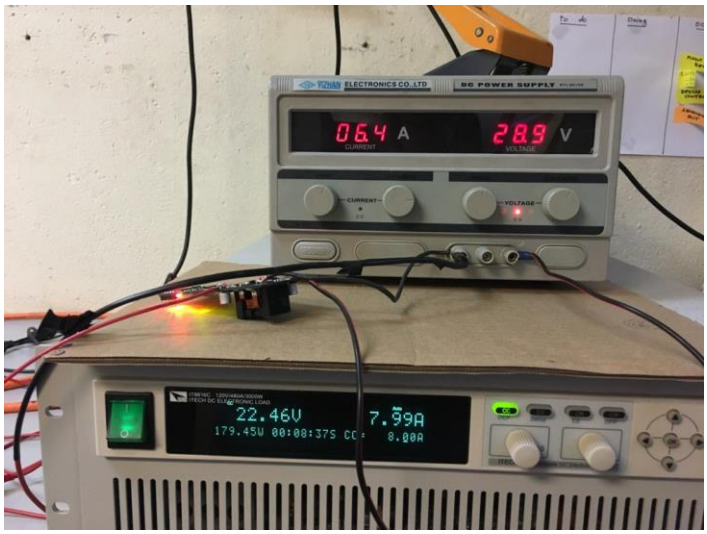


Figure 4.3: 24V voltage regulator real results

None of them suffered of an excessive overheating, reaching the 38ºC (LM3477, in 50% of the maximum power handled by the controller, 3A) and 32ºC (LM3477, in 50% of the maximum power handled by the controller, 10A). Both of these temperature measures were performed without any king of cooling system, that's why, the team considered these fact as a success. As a result, both voltage regulators ensue in a constant output voltage sources with low voltage ripples (lower than 1%) and relatively low current ripples (lower than 30%). The voltage and current ripples differ from the ones obtained in the simulations due to the usage of different inductors and capacitors which helped to reduce even more the current and voltage ripples, respectively.

As it was already mentioned, the exceptional situation lived during these last months caused important delays in the electronics' assembly and installation. The whole project itself is not finished because of the lack of testing inside the car. Currently, the team is working in finishing the whole electronic package of the car: wiring, ECUs, HV

wiring; and other aspects as testing and installing the new brake cylinder. Once the electronic unit is completed, the team will test it first outside the car, and then, inside of it.

Finally, with all the hardware and electronics installed inside the car, the team will carry out the testing of the whole car in the Montmeló circuit to check its correct functioning and search for possible problems.

5. Budget

This project budget could be divided in the designing and simulation phase, the PCB and the components used to manufacture the Rear ECU and the DC-DC voltage regulators cost.

Design & simulation

The design phase involved the software used to develop the Rear ECU schematics and the PCB layout (Altium), the Rear ECU microcontroller programming and simulation (CubeMX, Kvaser and Keil), and the simulation of the DC-DC regulators (Webench TI); and the hours implied in the designing phase. Most of this software is acquired by free licenses or by free keys handled by the company thanks to some sponsorship.

#	Resource	Provider	Cost	Observations
1	Quim Pérez Jiménez	Driverless UPC	0 €/h	Expected working time: 224h
2	Altium license	Altium	0 €	Acquired License (sponsorship)
3	CUBEMX software	CubeMX	0 €	Free License
4	Keil Software	Keil	0 €	Free License
5	Kvaser software	Kvaser	0 €	Acquired License (sponsorship)
6	Kvaser tool	Kvaser	0 €	Free Acquired (sponsorship)
7	Webench TI tool	TI	0 €	Free License
		TOTAL	**0 €**	

Table 5.1: Design & Simulation Budget

PCB manufacturing

The Rear ECU PCB manufacturing consists of the PCB itself, the components installed on it and the hours involved in its fabrication. The PCB was provided by EUROCIRCUITS manufacturer and the components were soldered in the UPC installations using its tools.

#	Resource	Provider	Observations
1	Quim Pérez Jiménez	Driverless UPC	Expected working time: 76h
2	4 layer PCB	Eurocircuits	Free Acquired (sponsorship)
3	Tin	UPC	Free Acquired
4	Flux	UPC	Free Acquired

5	Soldering station	UPC	Free Acquired
6	Components	Mouser & RS	Free Acquired (sponsorship &Driverless UPC budget)
		TOTAL	

Table 5.2: PCB manufacturing Budget

A more detailed list of the components used is shown in the CBOM (Costed Bill of Materials) of the Rear ECU attached in the appendices. The total cost of the ECU is shown at the end and it is 378.57 €. It must be highlighted that RS components and LEMO connectors were acquired through a sponsorship with their respectively companies, which means that in the reality these products were acquired for free.

DC-DC Regulator

[7] An estimation of the DC-DC regulator was done to check its worth and cost in front of commercial regulators:

LM3477

#	**Resource**	**Provider**	**Unit cost**	**Quantity**	**Cost**	**Obs.**
1	Ceramic Capacitors & Resistors	RS	0.3 €	12	3.6 €	Estimated cost
2	Electrolytic Capacitors	Mouser	1 €	2	2 €	Estimated cost
3	7443551920	RS	4.05 €	1	4.05 €	Inductor
4	BSC022N04LS6ATMA1	RS	1.27 €	1	1.27 €	Transistors
5	NSR0240HT1G	Mouser	0.21 €	1	0.21 €	Schottky Diode
6	LM3477MH/NOPB	RS	2.31 €	1	2.31 €	DC-DC controller
		TOTAL			**13.44€**	

Table 5.3: LM3477 estimated cost

LM25116

#	**Resource**	**Provider**	**Unit cost**	**Quantity**	**Cost**	**Obs.**
1	Ceramic Capacitors & Resistors	RS	0.3 €	16	4.8 €	Estimated cost
2	Electrolytic Capacitors	Mouser	1 €	2	2 €	Estimated

						cost
3	7443641000	RS	11.57 €	1	11.57 €	Inductor
4	BSC022N04LS6ATMA1	RS	1.27 €	2	2.54 €	Transistors
5	NSR0240HT1G	Mouser	0.21 €	1	0.21 €	Schottky Diode
6	LM25116MH/NOPB	RS	4.16 €	1	4.16 €	DC-DC controller
		TOTAL			**25.28€**	

Table 5.4: LM25116 estimated cost

6. Conclusions and future development:

In the end, after the whole the designing and manufacturing process, the Rear ECU was obtained with all the specifications and requirements established. It was reliable, able to supply the specified devices, acquire information from the sensors and communicate with other ECUs and devices from the car.

However, it was realised that the Rear ECU had room for improvements and it could be optimised. First, the inductor generation of EMIs could cause some interference in the control units as the microcontroller or the DC-DC voltage controller. For this reason, the inductor was installed in the opposite layer where the control units were located, using the ground layer as an EMIs isolator, but, it had been better to separate the supply section (with the DC-DC voltage regulator) and the microcontroller section (with the microcontroller, CAN communication, sensors, …) in two ECUs to obtain more reliability. Taking this idea into consideration and with reliable and high-power DC-DC regulators, what it could be considered is the designing of one or two ECUs equipped with switching regulators destined exclusively to transform the voltage from the LV battery to a stable source of 12V or 24V and supply all the ECUs and the devices of the car. This fact would avoid the installation of a voltage regulator in each ECU which would increase the reliability of the electronics in the car.

Moreover, some mistakes were found in the schematics design as the 5V wheel speed sensors supply instead of the 24V supply required. Another possible improvement could be the optimisation of the microcontroller code using semaphores.

As a result, the goal for the next year will be re-manufacture the Rear ECU taking into considerations the mentioned aspects, eliminating mistakes and increasing the reliability of the ECU. Furthermore, the team is looking to update the CAN communication protocol and replace it for the CAN-FD protocol, which provides faster data transfer, less number of undetected errors and an increase of the message payload size (64 bytes of data).

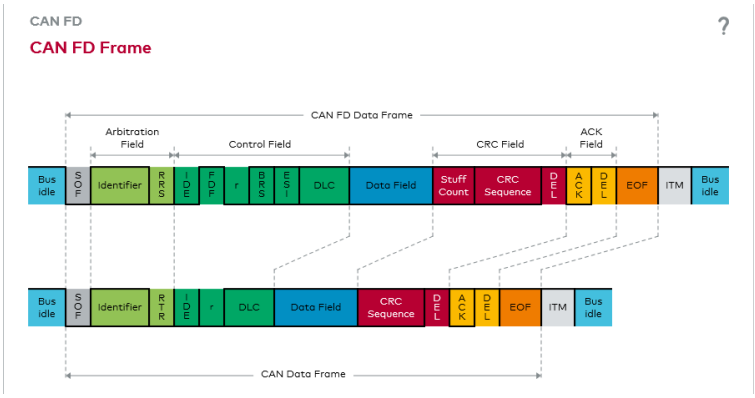


Figure 6.1: CAN FD

The DC-DC voltage regulators were satisfactory manufactured accomplishing all the established requirements and specifications: they were reliable and able to supply the specified currents with high efficiency (90-99%) and low overheating. The election of high quality transistors, with low internal on resistance and low parasitic gate charge, was the key to obtain the requirements established. Also, the high efficiency of the controller helped to reach the desired efficiency.

A future improvement could be the design of DC-DC voltage regulators using a microcontroller, which by a feedback signal and its programming, it would be able to control the duty cycle and the output voltage from the regulator. This would replace the commercial DC-DC voltage controller and it would result in a 100% customized voltage regulator which would provide more possibilities and freedom in the election of components and tuning the specifications of the voltage regulator.

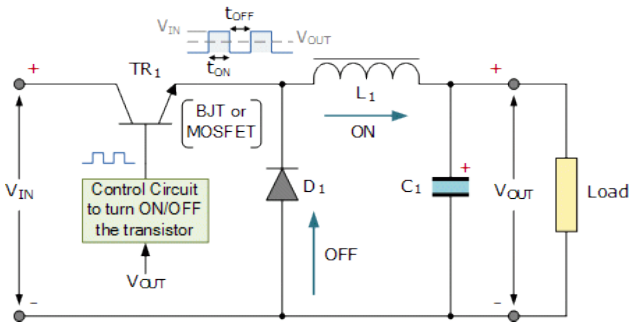


Figure 6.2: Switching regulator

Bibliography:

[1] Formula Student Germany 2020. [Online] Available: https://www.formulastudent.de/fsg/. [Accessed: September, October 2019 / March 2020]
[2] Altium 2020. [Online] Available: https://www.altium.com/solution/pcb-designing-tutorial/. [Accessed: October, November 2019]
[3] Texas Instruments 2013. [Online] Available: http://www.ti.com/lit/ds/symlink/lm3477.pdf?HQS=TI-null-null-digikeymode-df-pf-null-wwe&ts=1589969082406. [Accessed: October, November 2019 / April, May 2020]
[4] Texas Instruments 2016. [Online] Available: http://www.ti.com/lit/ds/symlink/lm25116.pdf. [Accessed: October, November 2019 / April, May 2020]
[5] Digikey 2020, [Online] Available: https://www.digikey.es/en. [Accessed: October, November 2019]
[6] Webench Texas Instrument 2020, [Online] Available: http://www.ti.com/design-resources/design-tools-simulation/webench-power-designer.html. [Accessed: December 2019 / 1, 2 May 2020]
[7] RS 2020, [Online] Available: https://es.rs-online.com/web/. [Accessed: December 2019 / January 2020]
[8] STMicroelectronics 2019. [Online] Available: https://www.st.com/resource/en/reference_manual/dm00031020-stm32f405-415-stm32f407-417-stm32f427-437-and-stm32f429-439-advanced-arm-based-32-bit-mcus-stmicroelectronics.pdf. [Accessed: February, March, April 2020]
[9] STMicroelectronics 2016. [Online] Available: https://www.st.com/resource/en/datasheet/dm00037051.pdf. [Accessed: February, March, April 2020]
[10] KVASER 2020, [Online] Available: https://www.kvaser.com/about-can/the-can-protocol/can-bit-timing/. [Accessed: February, 24 April 2020]
[11] ControllersTech 2019, [Online] Available: https://controllerstech.com/measure-pulse-width-using-input-capture-in-stm32/. [Accessed: February, 26 April 2020]
[12] E. Marinoni. “Tutorial on CUBE-MX and CUBE Library”. *EMCU*, 2019. [Online] Available: http://www.emcu.eu/cube-mx-and-cube-library/. [Accessed: February 2020]
[13] CBINSIGHTS 2020, [Online] Available: https://www.cbinsights.com/research/autonomous-driverless-vehicles-corporations-list/ . [Accessed: 27 & 28 March 2020]
[14] Auto connected car news 2020, [Online] Available: https://www.autoconnectedcar.com/adas-advanced-driver-assistance-sytems-definition-auto-connected-car/. [Accessed: 28 March 2020]
[15] Sophie Zoria. “Smart Cities: A New Look at the Autonomous-Vehicle Infrastructure”. *The Startup*, 2019. [Online] Available: https://medium.com/swlh/smart-cities-a-new-look-at-the-autonomous-vehicle-infrastructure-3e00cf3e93b2. [Accessed: 28 March 2020]
[16] Jim Hines. “Autonomous Vehicles are Driving Innovation”. *ElectronicDesign*, 2019. [Online] Available: https://www.electronicdesign.com/markets/automotive/article/21807786/autonomous-vehicles-are-driving-innovation. [Accessed: 29 March 2020]
[17] Bryce Johnstone. “Autonomous Vehicles: Are We There Yet?”. *ElectronicDesign*, 2020. [Online] Available: https://www.electronicdesign.com/markets/automotive/article/21122331/autonomous-vehicles-are-we-there-yet. [Accessed: 29 March 2020]
[18] Jaycar Electronics 2001, [Online] Available: https://wecanfigurethisout.org/ENERGY/Lecture_notes/Renewable_Distributed_Grid_Supporting_materials/dcdcconv.pdf. [Accessed: 1, 2, 27, 28 April 2020]
[19] Monzer Al Sakka, Joeri Van Mierlo and Hamid Gualous. “DC/DC Converters for Electric Vehicles”. *IntechOpen*, 2011. [Online] Available: https://www.intechopen.com/books/electric-vehicles-modelling-and-simulations/dc-dc-converters-for-electric-vehicles. [Accessed: 1, 2, 27, 28 April / 2020]

Glossary

A list of all acronyms and the meaning they stand for.

ECU Electronic Control Unit

DC Direct Current

PCB Printed Circuit Board

TS Tractive System (Systems destined to the movement of the car: motor, inverter …)

TSAL Tractive System Active Light (Light that shows if TS is activated or not)

HV High Voltage

LV Low Voltage

CAN Controller Area Network (Communication protocol used in automotive)

BOTS Brake Over-Travel Switch

IMD Insulation monitoring Device

BSPD Brake System Plausibility Device

AMS Accumulator Management System

SDC Shutdown Circuit (The shutdown circuit carries the LV to the whole car. It is defined as a series connection of at least two master switches, three shutdown buttons, the BOTS, the IMD, the inertia switch, the BSPD, all required interlocks and the AMS. If some of these sensors are activated the SDC opens and the car must stop).

WP Work Packages

FSG Formula Student Germany

FSC Formula Student Czech

FSS Formula Student Spain

AV Autonomous Vehicles

IoT Internet of Things

ADAS Advanced driver-assistance systems

MCU Microcontroller Unit

IC Integrated Circuits

BMS Battery Management System

HVD High Voltage Disconnect

ADC Analogic-to-Digital Converters

DMA Direct Memory Access

APB Advanced Microcontroller Bus

EBS Emergency Brake System

EMI Electromagnetic Interference